

TARTU ÜLIKOOL  
MATEMAATIKA-INFORMAATIKATEADUSKOND

Arvutiteaduse Instituut  
Infotehnoloogia eriala

Karl Kilgi

LEGO MINDSTORMS NXT'ga ühilduva prototüüpimislaua  
tutvustus ja ülesanded

Bakalaureusetöö (6 EAP)

Juhendaja: Anne Villems

Kaasjuhendaja: Taavi Duvín

Autor: ..... “.....” juuni 2011

Juhendaja: ..... “.....” juuni 2011

Kaasjuhendaja: ..... “.....” juuni 2011

Lubada kaitsmisele

Professor: ..... “.....” juuni 2011

TARTU 2011

# Sisukord

Sissejuhatus.....	3
1. Prototüüpimislaud.....	5
1.1 Prototüüpimine.....	5
1.2 Maketiplaan .....	5
1.3 NXT prototüüpimislaud.....	7
1.4 NXT prototüüpimislauda ühendused .....	9
1.5 Suhtlus NXT prototüüpimislauga.....	11
1.6 NXT prototüüpimislauda mikrokontroller .....	13
1.6.1 NXT prototüüpimislauda mikrokontrolleri protsessor.....	14
1.6.2 NXT prototüüpimislauda mikrokontrolleri mälu.....	14
2. NXT Prototüüpimislauda kasutamine.....	18
2.1 Seadmete ühendamise .....	18
2.2 NXT prototüüpimislauda programmeerimine keeles NXT-G .....	19
2.2.1 NXT-G prototüüpimislauda programmeerimisploki lisamine.....	19
2.2.2 NXT-G prototüüpimislauda programmeerimisploki kasutamine .....	20
2.3 NXT prototüüpimislauda programmeerimine keeles NXC .....	24
2.3.1 Programmeerimiskeskond Bricx Command Center.....	24
2.3.2 NXC Standardteek.....	25
2.3.3 Programmeerimine NXC's .....	25
3. Ülesanded .....	28
3.1 Ülesanne 1 - Vilkuv LED .....	29
3.2 Ülesanne 2 - Jooksvad tuled .....	32
3.3 Ülesanne 3 - Valguse andur.....	35
3.4 Ülesanne 4 - Reaktsiooni aja mõõtmine .....	39
Kokkuvõte .....	43
Abstract.....	44
Kasutatud kirjandus.....	45
Lisad .....	48
Lisa 1 – Ülesannete lahendused .....	48

# Sissejuhatus

Õppida valdkonda viisil, millel puudub eluline ja praktiline pool, on suhteliselt raske. Enamjaolt puudutab liigne teoreetilisus reaallaineid. See võõrandab noori reaallainetest, kuid üha enam kasvab nõudlus just selle valdkonna spetsialistide järele. Tegelikult on olemas vahendeid, mille abil saab reaallainete õpetamist huvitavamaks muuta. Kasutades tänapäeva tehnoloogiat, mis võimaldab reaallaineid esitada motiveerivamalt ja kaasahaaravamalt, saab noortes nende vastu tekitada suuremat huvi.

Aastal 2006 tuli välja LEGO MINDSTORM NXT komplekt, mille kasutamine tunnitööks on üks võimalus reaallaineid huvitavamalt õpetada. Tänapäevaks on LEGO robotid kasutuses kõigis kooliastmetes. Algkoolides on robotitega tegelemine küll rohkem mängulisem, kuid põhikoolis ja gümnaasiumis viiakse nende abil läbi keemia, matemaatika, füüsika, informaatika ja palju teisi tunde.

Alates 2007. aastast viiakse Eesti koolides läbi Kooliroboti projekti [1], mille raames on õppeasutustel võimalik saada endale LEGO MINDSTORMS NXT komplekte. See on õppevahend, mis koosneb NXT põhiplokist, mootoritest, anduritest, vajalikest juhtmetest ja LEGO klotsidest. Kuna komplektis on ainult piiratud hulk andureid, siis on hakanud firmad nagu Vernier[2], HiTechnic[3], Mindsensors[4] ja Codatex[5] tootma LEGO MINDSTORMS NXT'ga ühilduvaid andureid. Antud töös kirjeldatakse lähemalt ettevõtte HiTechnic poolt loodud prototüüpimislauda.

Kahjuks on kolmandate osapoolte valmistatud andurite dokumentatsioon sageli lünklik ning sellekohane eestikeelne informatsioon puudub sootuks, mistõttu on nende andurite kasutamine Kooliprojekti raames raskendatud. Töö eesmärgiks on anda lugejale põhjalik ülevaade NXT prototüüpimislaust ning vältimaks selle liiga teoreetiliseks muutumist, on koostatud ka seda tutvustavad praktilised ülesanded.

Käesolev bakalaureusetöö koosneb kolmest suuremast osast:

- Esimeses osas tutvustatakse prototüüpimise põhimõtteid ning antakse ülevaade NXT prototüüpimislauda tehnilistest parameetritest koos selgitustega miks neid vaja on ja kus kasutada saab.

- Teine osa keskendub NXT prototüüpimislaua kasutamiseks vajalike programmide valmistamisele NXT-G ja NXC keeles. Samuti antakse nõuandeid ning soovitusi NXT prototüüpimislaua ohutuks kasutamiseks.
- Kolmandas osas on neli erineva raskusastmega ülesannet, mida on võimalik lahendada, kasutades NXT prototüüpimislauda ja lihtsamaid vooluringi komponente. Lahendamiseks peab ka programmeerima NXT-G või NXC keeles. Ülesannete lahendusprogrammid on CD'l, mis asub Lisas 1.

Antud bakalaaurusetöös esitatud materjali koostamisel on silmas peetud asjaolu, et see oleks üheselt ja lihtsalt arusaadav nii õpetajatele kui ka õpilastele.

# 1. Prototüüpimislaud

Enne, kui saab alustada töös uuritava prototüüpimislaua tutvustamist, tuleks kõigepealt selgeks teha selle kasutamiseks vajalikud komponendid ning mõista prototüüpimise olemust. Järgnevalt antakse ülevaade maketiplaanist, vooluahelast ning prototüüpimise põhimõtetest üldisemalt.

## 1.1 Prototüüpimine

Prototüüpimine on tegevus, mis aitab toodet luua ja arendada. Tähtis on märkida, et prototüüpimisel ja selle väljundis ei ole alati kasutusel lõpptootes kajastuvad komponendid, vaid osad, mida oleks võimalikult lihtne muuta ja testida. Seda ütleb ka kreeakeelne sõna *πρωτότυπον* (*prototypon*) ehk primitiivne vorm millestki [6]. See kõik annab võimaluse arendajale näha lihtsamini vigu ning otsida alternatiivseid lahendusi.

Elektroonikas tähendab prototüüpimine vooluahela loomist, et teada saada, kas see töötab nii, nagu ette nähtud. Prototüübi loomisel kasutatakse näiteks maketiplaani, juhtmeid, transistoreid, erinevaid takisteid ja prototüüpimislauda [7]. Prototüüpimislaud annab võimaluse lihtsustada arenduse käiku, sisaldades erinevaid komponente, mis on vajalikud ühe elektroonilise toote tööks. See tähendab, et kasutaja eest on juba osa tööd tehtud ning keskenduda saab toote arendamisele ja testimisele.

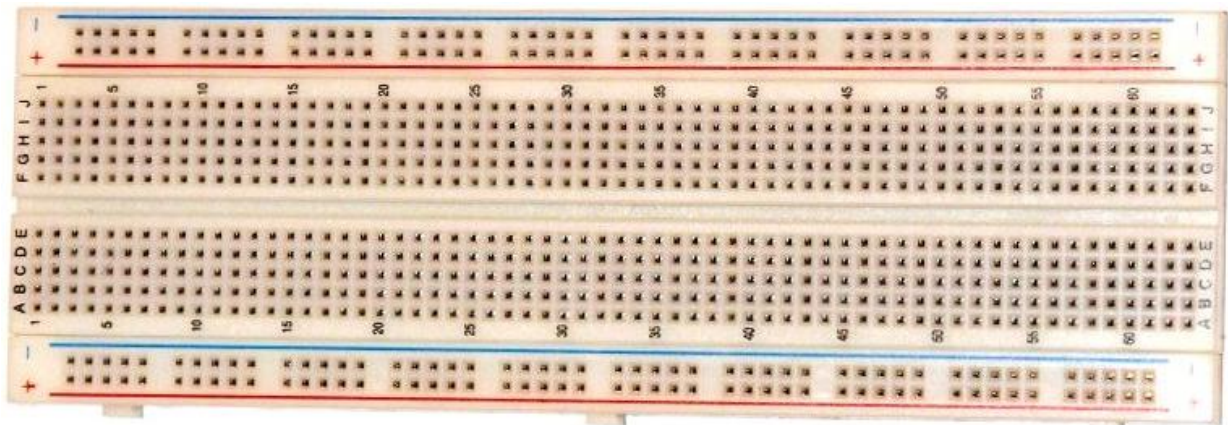
Prototüüpimislaua puhul on oluline, mis võimalusi see arendajale pakub, kuna olenevalt arendatava toote keerukusest ja vajadustest võib prototüüpimislaud väikseks jääda. Käesolevas töös kasutatav prototüüpimislaud firmalt HiTechnic on loodud küll hariduslikul eesmärgil, kuid selle abil on võimalik aru saada, kuidas ühe elektroonilise toote arendamine käib. Seega algajal ei tasu muretseda suurt prototüüpimislauda, sest mida rohkem võimalusi, seda raskem on prototüüpimislaua käitumist mõista ning seda kõrgem on ka selle hind.

## 1.2 Maketiplaan

Skeeme saab prototüüpimislauga luua kahel erineval viisil, kas neid kinni jootes või kasutades selleks spetsiaalset vahendit, maketiplaani. Tavaliselt kasutatakse jootmist alles siis, kui kõik on läbi testitud ning tulemus on püsiv, vastasel juhul on vajalikuks tööriistaks

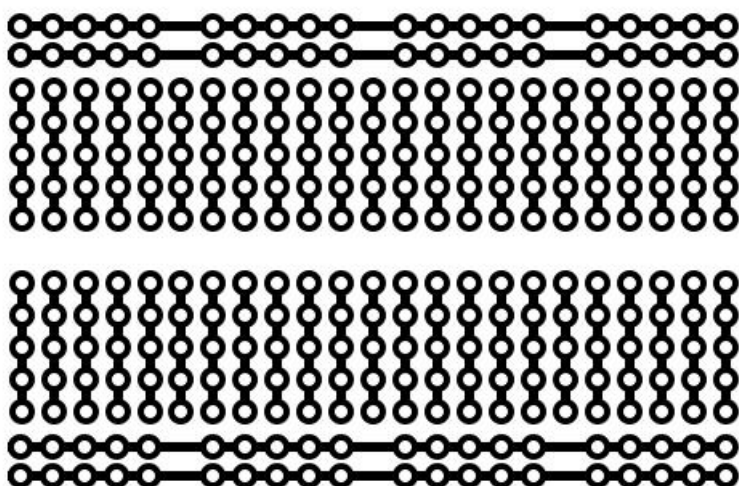
maketiplaan. See annab võimaluse hoida ahelat koos ning ühendada omavahel kõik vajalikud komponendid. Maketiplaan koosneb blokkide kaupa augustatud plastiktükist (vt. Joonis 1), mille keskel on kas fosforpronksi või uushõbeda ehk nikli, vase ja raua sulamist moodustatud skeem (vt. Joonis 2). Maketiplaanil paiknevate aukude vahe on 2.54mm ning sellele suurusele vastab ka prototüüpimislaua sisend/väljundotsade paigutus [8].

Tüüpiline asetus koosneb kahest põhilisest alast, mida nimetatakse ribadeks, nendes omakorda paiknevad sisemiselt ühendatud skeemid. Esimese, keskel asuva riba ülesandeks on hoida ning ühendada elektroonilisi komponente. Riba keskel on sälk, mis jagab selle kaheks võrdseks blokiks. Joonisel 2 on näha kuidas seal paiknevad ühendused jooksevad vertikaalselt. Selle riba esimese bloki ots märgistatakse tähtedega “a b c d e” ning teise bloki ots tähtedega “f g h i j”. Olenevalt tootjast, võib otsmine märgistus olla ka trükitähtedes [8]. Tavaliselt on maketiplaani hõlpsamaks kasutamiseks märgitud keskmise bloki külgedele numbrid, mis on HiTechnic’u prototüüpimiskomplekti kuuluva maketiplaani puhul iga 5. augu juures.



**Joonis 1.** *Maketiplaan* [9]

Teine riba koosneb kasutuses oleva maketiplaani puhul samuti kahest blokist ning need on paigutatud äärtesse. Sellel blokil on kaks ülesannet. Esimese osa, mille otstele on märgitud “+” ning äärde punane joon, ülesandeks on voolu jagamine seadmetele üle maketiplaani. Teise osa, mille otstel “-” märk ning kõrval sinine joon, tööks on vooluahela negatiivse väljundi “kokku korjamine” ning selle miinus клеммиле lähemale viimine.



**Joonis 2.** *Maketiplaani skeem*

Maketiplaanil kasutatakse ühenduste loomiseks näiteks erinevaid juhtmeid, takisteid. HiTechnic’u prototüüpimiskomplektiga tuleb kaasa varieeruvate pikkustega 70 eelpainutatud juhet [10]. Need on kooritud otstega ning ei vaja maketiplaaniga ühendamiseks spetsiaalset töötlust. Seega on maketiplaan kergesti taaskasutatav, mis tähendab, et sellega on hea eksperimenteerida. Ilma erilise vaevata saab komponendid ümber tõsta või sisestada indikaatori, näitamaks mõne osa seisu.

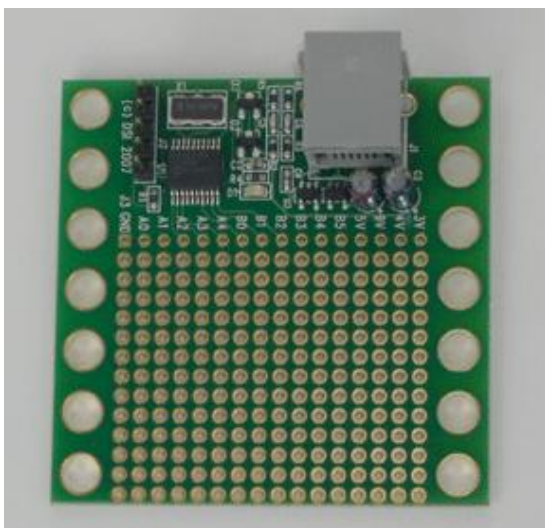
### **1.3 NXT prototüüpimislaud**

Firma HiTechnic poolt loodud prototüüpimislaud (vt Joonis 3), mis ühildub LEGO MINDSTORMS NXT’ga (edaspidi NXT prototüüpimislaud), on oma võimaluste poolest hea vahend algajale arendajale, kuid samas pakub huvitavaid lahendusi ka edasijõudnutele. LEGO MINDSTORMS NXT annab sellega kasutajale võimaluse ise oma vajaduste järgi andureid või hoopis midagi muud huvitavat ehitada [10]. Võimalused on lõputud, näiteks saab sellega ehitada endale värvide tuvastaja, panna LED tuled oma soovi järgi vilkuma või hoopis valmistada seadme, mis suudab toas olevat temperatuuri mõõta.



**Joonis 3.** Jooteta NXT prototüüpimislaud [10]

NXT prototüüpimislaudu on olemas kahte eri tüüpi: jooteta laud, mille kasutamiseks on vajalik maketiplaani ja joodetav laud, mille kasutamiseks tuleb detailid prototüüpimisalasse kinni joota (vt Joonis 4). Tehniliste omaduste poolest on need kaks täiesti sarnased.



**Joonis 4.** Joodetav NXT prototüüpimislaud [11]

Selles töös on autoril kasutamiseks jooteta NXT prototüüpimislaud koos maketiplaaniga. Vajadusel on võimalik joota ka jooteta prototüüpimislauale ala, kus saab oma komponente jäädavalt hoida. Järgnevatel peatükkidel antakse ülevaade NXT prototüüpimislauda tähtsamatest omadustest, et mõista selle toimimist ja kasutamise võimalusi erinevates olukordades.



## 1.4 NXT prototüüpimislaua ühendused

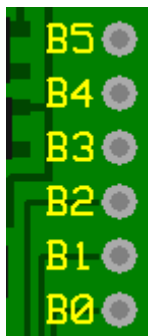
Elektroonilise komponendi töötamiseks on vaja elektrit, kuid seda saab kasutada ka vahendina, mis aitab meil informatsiooni edastada. Me vajame NXT prototüüpimislaua paiknevad ühendusi, et ehitada erinevaid andureid, seadmeid. Käesolev punkt annab ülevaate NXT prototüüpimislaua ühendustest koos mõningate selgitustega. Tehnilised andmed pärinevad allikast [10].

NXT prototüüpimislaua on 3.3v, 4.3v, 5.0v ja 9.0v toiteühendused (vt. Joonis 5), kus 3.3v ja 5.0v on pingeregulaatoriga reguleeritud vastavalt reguleerimata pingetest 4.3v ja 9.0v. Maksimaalne voolutugevus on 3.3 ja 4.3v korral 20mA ning 5.0v ja 9.0v puhul 12mA. Neid ühendusi kasutatakse tavaliselt seadmete jaoks, mis vajavad oma tööks alalisvoolu.



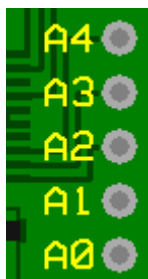
**Joonis 5.** NXT prototüüpimislaua paiknevad toiteühendused [10]

Kuus digitaalset sisend/väljundühendust B0 - B5 (vt. Joonis 6), mida saab vastavalt vajadusele konfigureerida kas kirjutama või lugema. Digitaalseid ühendusi kasutatakse seadmete puhul, kus läheb vaja tööks diskreetseid väärtusi [12]. Siin on kasutusel kahendsüsteem ehk kas ühenduses on pinge või mitte. Oletame, et tahame saata NXT prototüüpimislauale morsekoodis sõnumit, siis peab olema lüliti, millele vajutades tekib pinge. Pinge peab muutuma 0 ja 3.3v vahel, kuna lugemise puhul on kahendsüsteemiväärtus 0 võrdne pingete 0v - 0.5v korral, 1 aga pingete vahemikus 2.0v- 3.3v. Järgnevalt tuleb teha programm, mis peaks aru saama, kui kaua digitaalsisendis pinge on - kas morsekoodi tuleb lisada lühike või pikk signaal. Sellest on omakorda võimalik tähti, numbreid ja sümboleid välja lugeda. Kui on tegemist digitaalse kirjutamisega, siis NXT prototüüpimislaua on ise pingete muutja, ning 0 väärtuse puhul on digitaalväljundis pinge 0v ning 1 korral 3.3v.



**Joonis 6.** *NXT prototüüpimislaua paiknevad digitaalsed ühendused* [10]

Viis analoogsisendit A0 - A4 (vt. Joonis 7), millel taga on 10-bitine muundur, mis muudab analoogsisendi digitaalseks. Sisendid on mõeldud kasutamiseks 0-3.3v pingega ning selle konverteerib muundur vahemikku 0-1023. Saadud arvu tagasi voltideks arvutamine käib valemi kaudu  $V = X * 129 / 40000$ , kus X tähistab saadud arvu vahemikus 0-1023 [10]. Näiteks võib analoogsisendit rakendada temperatuurianduri ehtamiseks, kasutades termistorit [13] (takisti, mis muudab oma takistust märgatavalt temperatuuri muutuse korral) ja toiteühendust. Analooģsisendis läbi toiteühenduse ja termistori jõudev pinge sõltub temperatuurist ning sellest saab järeldada, kui soe või külm on.



**Joonis 7.** *NXT prototüüpimislaua paiknevad analoogsisendid* [10]

Toiteallika miinusklemm (vt. Joonis 8), mis on ühenduses NXT prototüüpimislaua toiteallika negatiivse poolega. Sinna peaksid jõudma kõik negatiivsed väljundid, näiteks LED tule puhul peab sinna ühendatud olema selle katoode.

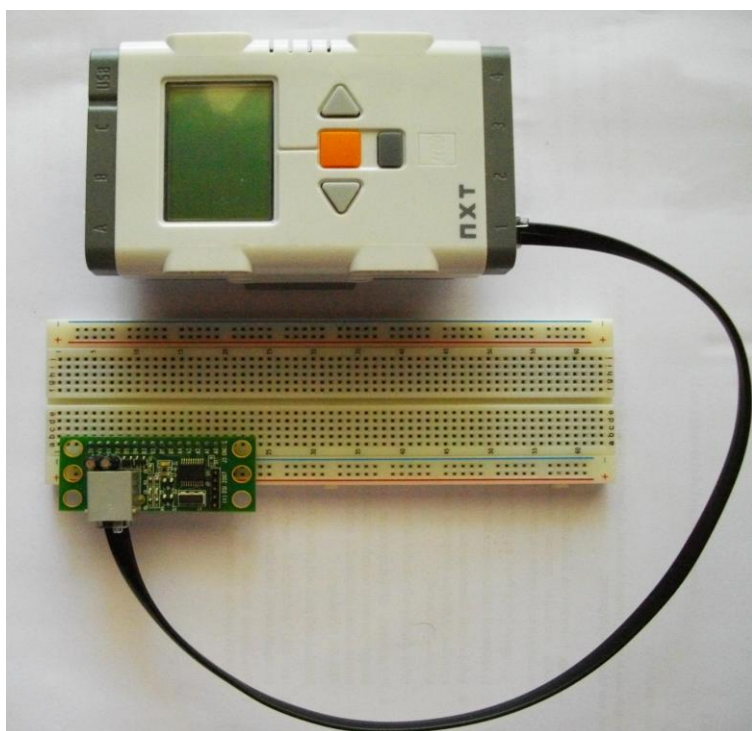


**Joonis 8.** *NXT prototüüpimislaua paiknev miinusklemm*

Nende ühenduste kasutamine andurite ja seadmete ehitamiseks on ainult pool tööst. Järgnevalt tulevad selgitused NXT prototüüpimislaua kasutuse ja suhtluse ning ühendusi juhtiva mikrokontrolleri kohta.

## 1.5 Suhtlus NXT prototüüpimislauga

NXT prototüüpimislaua kasutamiseks on vaja NXT põhiplokki. Sellega ollakse ühenduses läbi kaabli, mis on ühendatud põhiploki standardsesse NXT sensoriporti (vt. Joonis 9). Põhiploki läheb vaja NXT prototüüpimislaualt andmete kättesaamiseks ning sinna saatmiseks. Näiteks peab analoogsisendi saamiseks NXT prototüüpimislaualt põhiplokk tegema päringu kindlasse registrisse, kuhu prototüüpimislaud selle paigutanud on. Tahtmisel midagi digitaalväljundisse kirjutada, peab põhiplokk saatma andmeid, millega oskab NXT prototüüpimislaud oma digitaalühendusi juhtida.



**Joonis 9.** Põhiplokiga ühendatud NXT prototüüpimislaud maketiplaanel

Selleks, et NXT prototüüpimislaud ja põhiplokk saaksid omavahel suhelda, tuleb esmalt kokku leppida reeglites. Seda nimetatakse suhtlusprotokolliks, mis koosneb formaalsest kirjeldusest, kuidas peaksid välja nägema digitaalsete sõnumite formaat ning reeglitest, mida tuleb järgida nende sõnumite vahetamisel. Protokollist on võimalik välja lugeda suhtluse süntaksit, semantikat ning ka sünkronisatsiooni [14]. NXT Prototüüpimislaud kasutab

suhtluseks põhiplokiga I<sup>2</sup>C suhtlusprotokoll. Järgnev info I<sup>2</sup>C protokoll kohta pärineb allikast [15].

I<sup>2</sup>C on tööstuslik kommunikatsioonistandard, mille arendas välja Taani elektroonikagigant Philips 1980ndate aastate alguses. Sellest ajast saadik on seda kasutatud erinevates tööstuskomponentides ja kohtades, kus on vajalik läbi viia lihtsat digitaalset suhtlust. I<sup>2</sup>C näol tegemist sõnumsideprotokolliga, kus üksteisega suhtlevad seadmed vahetavad omavahel lühikesi sõnumeid. Seda kasutatakse tavaliselt aeglaste lisaseadmete ühendamiseks mõne emaplaadi, manussüsteemi või mobiiltelefoniga.

LEGO NXT juhtplokil on selliseid andmesiine neli – üks iga anduripordi jaoks [16]. Kuna ühe siini külge võib olla korraga ühendatud palju seadmeid, tuleb neid kuidagi üksteisest eristada. Selleks on igal seadmel unikaalne 7-bitine aadress ning seadmed vastavad ainult nendele sõnumitele, mis on neile adresseeritud. 7-biti maksimaalne väärtus kümnendsüsteemis on 127, arvestades ka 0 väärtust saab kokku 128 erinevat võimalust. Seega on võimalik ühendada kuni 129 erinevat seadet, kus üks neist on ülema rollis (*master*) ning ülejäänud alama rollis (*slave*). Ainult ülema rollis olev seade saab algatada suhtlust. See saab saata alamatele käsked ning pärida nendelt andmeid, alamad saavad ainult vastata, nemad ise ei saa I<sup>2</sup>C siinil suhtlust algatada. NXT juhtplokk saab olla ainult ülema rollis, mis tähendab, et ta peab alati ise algatama ning juhtima sideseansse iga andmesiini peal.

NXT juhtplokk edastab anduritele kahte tüüpi sõnumeid:

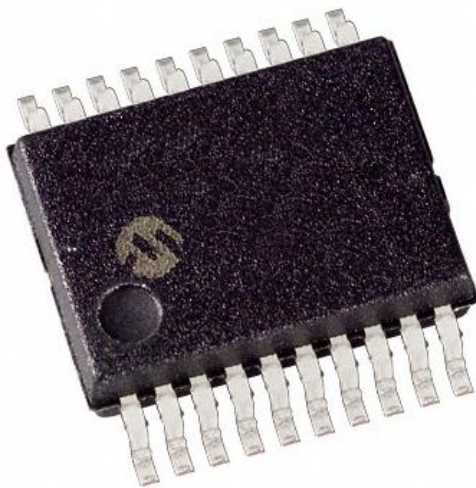
- "kirjuta" (ingl.k. write) tüüpi sõnumiga saadetakse üks või mitu baiti andmeid,
- "loe" (ingl.k. read) tüüpi sõnumite abil loetakse üks või mitu baiti andmeid.

Mõlemad sõnumitüübid algavad päisega, kus on kirjas selle anduri aadress, millele sõnum on mõeldud, ja anduri sisemine register, kuhu tahetakse midagi kirjutada või kust midagi lugeda. Sellest registrist võib mõelda kui ühest anduri sees olevast mälupesast. Igal registril on oma kindel roll ning anduriga suhtlemine ja isegi selle töö juhtimine seisnebki selles, et vastavatesse registritesse kas kirjutatakse andmeid või loetakse seal juba olevaid andmeid. Peatükis 1.6.2 on tabel, mis sisaldab NXT prototüüpimislaua oluliste registrite aadresse ning lühikest kirjeldust, mida mingi register teeb.

Iga LEGO NXT programmeerimiskeskonna jaoks eksisteerivad vastavad vahendid I<sup>2</sup>C suhtluse läbiviimiseks. Sellist madalatasemelise suhtluse programmeerimist läheb vaja NXC keeles. NXT-G keeles on olemas aga mugavusfunktsioonid, mis peidab kasutaja eest suhtlusprotokolli enda keerukuse.

## 1.6 NXT prototüüpimislaua mikrokontroller

Prototüüpimislauadel on palju olulisi komponente, kuid üheks kõige tähtsamaks võib lugeda mikrokontrollerit. Võib öelda, et mikrokontroller on omamoodi väike arvuti. Sellel on oma protsessor, mälu ning programmeeritavad sisend/väljundseadmed [17]. NXT prototüüpimislaua paikneva mikrokontrolleri tootjaks on Microchip Technology ning see kannab nime PIC16LF819-I/SS [10]. Ülevaate sellest, milline näeb välja see mikrokontroller, annab joonis 10.



**Joonis 10.** Mikrokontroller PIC16LF819-I/SS (ilma seerianumbriga) [18]

Mikrokontrolleritel on erinevad pakendamisviisid, millest sõltub näiteks mitme jalaga see on või kuidas see on ühendatud ülejäänud prototüüpimislauaga. Selle mikrokontrolleri pakendamisviisi kood on 20SSOP. See tähendab, et sellel kasutatakse ühendusteks 20 jalga ning SSOP - *Shrink small-outline package* pakendusviisi. SSOP kiipide jalgade vahe on 0.635mm ning need asetsevad kahel pikemal äärel [19].

### 1.6.1 NXT prototüüpimislaua mikrokontrolleri protsessor

Eelmainitud mikrokontrolleri peal asuva väikse protsessori kaks kõige tähtsamat näitajat on tema tuuma suurus ning taktsagedus. Suuremate protsessorite puhul, näiteks arvutis paiknevatel, on neid tegureid muidugi palju rohkem. Kuid need kaks näitajat annavad antud juhul teada, kui kiiresti võimaldab see protsessor näiteks arvutada, andmeid lugeda, kirjutada ning sisendeid digiteerida. Sellel mikrokontrolleril asuva protsessori tuum on 8-bitine ning maksimum taktsagedus on 20MHz [20].

Protsessor töötab kasutades mälus asuvat püsivara, mis on kirjutatud HiTechnic'u enda loodud *assembleris*, mille vahendusel viiakse läbi kõik kasutaja programmeeritud toimingud, mis on edastatud 8 biti kaupa protsessori tuuma. See on suletud lähtekoodiga, seega ei ole kasutajal võimalik seda ise uuendada või oma soovi järgi muuta. Seda on võimalik küll üle kirjutada uue püsivaraga, kasutades selleks spetsiaalseid vahendeid, kuid sellisel juhul kaotab NXT prototüüpimislaud tootjapoolse garantii.

### 1.6.2 NXT prototüüpimislaua mikrokontrolleri mälu

Mikrokontrolleril asub kolm erinevat mälu [20]:

- Programmeeritava väikmälu ehk *Flash memory* suuruseks on 3584 baiti. See mälu hoiab oma informatsiooni alles ka pärast toite kadumist, mis tähendab, et seal hoitakse protsessori tööks vajalikku püsivara ning informatsiooni sensori kohta - sensori versiooni number, tootja, sensori tüüp (vt. Tabel 1).
- Elektriliselt kustutatava ja programmeeritava püsimälu ehk EEPROM'i suuruseks on 256 baiti. Selles mälus hoitakse tavaliselt algseadistusi, kuid NXT prototüüpimislaual need puuduvad, seega EEPROM mälu ei kasutata.
- Staatilise muutmälu ehk SRAM'i suurus on 256 baiti. Seal hoitakse jooksvat informatsiooni, mille puhul ei ole oluline selle säilimine pikemas perspektiivis.

Järgnevas tabelis on kirjeldatud NXT prototüüpimislaua mälujaotus, mille poole pöördumine käib I<sup>2</sup>C suhtlusprotokolli aadresside kaudu:

Aadress	Tüüp	Väli
00 – 07H	Sümbolid	Sensori versiooni number
08 – 0FH	Sümbolid	Tootja
10 – 17H	Sümbolid	Sensori tüüp
42H	Bait	A0 ülemised 8 bitti
43H	Bait	A0 alumised 2 bitti
44H	Bait	A1 ülemised 8 bitti
45H	Bait	A1 alumised 2 bitti
46H	Bait	A2 ülemised 8 bitti
47H	Bait	A2 alumised 2 bitti
48H	Bait	A3 ülemised 8 bitti
49H	Bait	A3 alumised 2 bitti
4AH	Bait	A4 ülemised 8 bitti
4BH	Bait	A4 alumised 2 bitti
4CH	Bait	Digitaalsed sisendid (bitid 0 – 5)
4DH	Bait	Digitaalsed väljundid (bitid 0 – 5)
4EH	Bait	Digitaalse sisend/väljundi kontroll (bitid 0–5)
4FH	Bait	Näidu võtmise aeg

**Tabel 1.** NXT Prototüüpimislaua mälu jaotus [10]

Järgnevalt on selgitatud tabelis antud mäluväljade väärtused:

- Sensori versiooni numbri väli tagastab NXT prototüüpimislaua versiooni numbri kujul “Vn.m”, kus n tähistab põhiversiooni ning m selle versiooni revisjoni numbrit. Versiooni numbri teadmisega saab otsida spetsiifilisi iseärasusi, eripärasid, miks midagi ei tööta nii, nagu kusagil on kirjeldatud, või ei tööta mõni programm nii nagu vanema versiooni peal. Töö autoril on kasutuses versioon numbriga 1.0, mis tähendab, et programmid, mis selles töös valmistatakse, ei pruugi tööle minna samamoodi ka uuema NXT prototüüpimislaua versiooniga.
- Tootja väljal paikneb tekst “HiTechnc”, mis on see firma, kes on antud sensori tootja. Väljad LEGO sensoritel on 8 baiti, sest see on maksimaalne, mida LEGO NXT blokk suudab korraga lugeda. Sellest tulenevalt on tootja nimi “HiTechnic” lühendatud kujule “HiTechnc”. Kõik HiTechnic PC suhtlusprotokolliga sensorid kasutavad sellisel kujul tootja nime.
- Sensori tüübi väljal paikneb “Proto”, mis näitab ära, mis sensoriga tegu on.

Kuna NXT prototüüpimislaua on 10-bitine analoog - digitaalmuundur, kuid mikrokontrolleril asuv protsessor ning LEGO NXT blokk toetavad korraga 8 biti käsitlemist, siis jagatakse need bitid mäluväljadel vastavalt:

- A0 - A4 ülemiste 8 biti väljadel tagastatakse hetke viie analoogsisendi ülemised 8 bitti.
- A0 - A4 alumiste 2 biti väljadel tagastatakse hetke viie analoogsisendi alumised 2 bitti.

Digitaalsete ühendusi on 6, seega on nende kasutamiseks vajalik ainult 6 bitti ehk bitid kohtadel 6 ja 7 on alati määratud nulliks. Digitaalsed sisendite jaotus nendel bittidel on kirjeldatud Tabelis 2.

- Digitaalsete sisendite väljad tagastavad kuue digitaalse signaali väärtuse bittidel 0-5.
- Digitaalsete väljundite väljal võidakse määrata mis tahes kuue digitaalse signaali väärtused, mis on konfigureeritud kui väljundi väli.
- Digitaalse sisend/väljundi kontrolli väli võimaldab konfigureerida kuue digitaalse signaali suuna bittidel 0-5. Bitt, mille väärtuseks on 0, määrab ära, et tegemist on sisendiga, ning 1, et väljundiga.



Ühendus	Väärtus
B0	01
B1	02
B2	04
B3	08
B4	16
B5	32

**Tabel 2.** Digitaalsete ühenduste jaotus [10]

- Näidu võtmise aja väli määrab kahe näidu vahelise võtmise aega, teisisõnu, kui tihti uuendatakse seis. Seda on võimalik määrata skaalas 4ms kuni 100ms.

Nüüdseks oleme tutvunud prototüüpimislaua ühenduste, nende kasutusvõimaluste ning elementidega, mis neid juhivad. Järgnevas peatükis selgitatakse NXT prototüüpimislaua ohutut käsitlust maketiplaani ja selle kasutamiseks vajalike programmide valmistamist.

## 2. NXT Prototüüpimislaua kasutamine

Järgnev peatükk kirjeldab NXT prototüüpimislaua kasutamist maketiplaanil ning selle programmeerimist keeles NXT-G. Mõistmaks paremini NXT prototüüpimislaua töötamist ning PC suhtlusprotokolli tööd on kirjeldatud ka selle programmeerimine natukene madalamas keeles Not eXactly C (edaspidi NXC) [21].

### 2.1 Seadmete ühendamine

Hoiatus! Prototüüpimislauda ja teisi elektroonilisi komponente võib kahjustada staatiline elektrilaeng. Nendega tuleb ettevaatlikult ringi käia ning hoiduda staatilistest laengutest, soovituslikult kasutades antistaatilisest randmepaela (vt. Joonis 11).



**Joonis 11.** *Antistaatiline randmepael* [22]

Tavaliselt on esimeseks teoks prototüüpimislaua kasutamisel selle maketiplaaniga ühendamine. Seda tuleks teha, hoides kinni prototüüpimislaua äärtest, surudes selle all asuvad ühendusteks kasutatavad klemmid maketiplaani. Uue maketiplaani puhul võib see nõuda rohkem jõudu, kuid alati on tähtis see, et prototüüpimislauda surutakse otse alla ning selle klemmid paiknevad kohakuti maketiplaani aukudega.

Kasutades prototüüpimislauda seadmete ja andurite ehitamisel, tuleks alati kontrollida nende komponentide sobivust. Ennem käivitamist tuleks korduvalt üle käia valmistatud vooluring, et hoiduda lühistest. Samuti tuleks üle vaadata pingete ja voolutugevuse vastavus, et mitte kahjustada nii prototüüpimislauda kui ka väliseid elemente. Näiteks liiga tugev pinge analoogsisendisse võib põhjustada prototüüpimislaua mikrokontrolleri läbipõlemise.

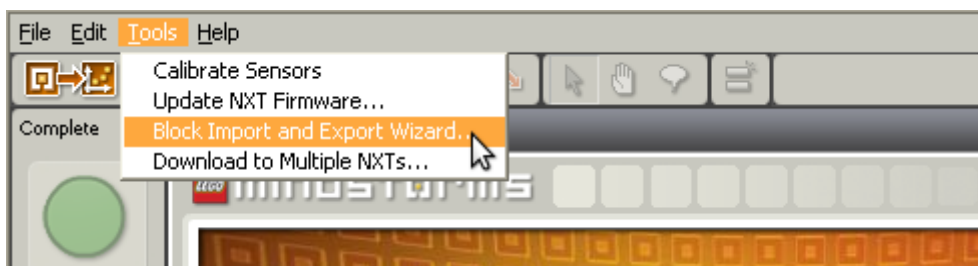
## 2.2 NXT prototüüpimislaua programmeerimine keeles NXT-G

NXT-G keel on graafiliselt programmeeritav, mis tähendab, et programmi valmistamiseks kasutatakse “lohistamistehnikat”. Sobib hästi esimeseks programmeerimiskeeleks, kuna hõlmab põhilisi programmeerimise aspekte ning on tänu graafilisele programmeerimiskeskkonnale lihtsasti arusaadav. NXT-G keele üheks suurimaks puuduseks võib lugeda piiratud keerukust, mis ei võimalda kompleksseid programme koostada või on need raskesti rakendatavad. Järgnevalt on õpetatud NXT prototüüpimislaua programmeerimisploki arenduskeskkonda lisamist ning selgitatud selle programmeerimise võimalusi.

### 2.2.1 NXT-G prototüüpimislaua programmeerimisploki lisamine

LEGO MINDSTORMS NXT 2.0 arenduskeskkonnas ei ole algselt olemas NXT prototüüpimislaua sensori plokki. See tuleb firma HiTechnic koduleheküljelt alla laadida ning programmi importida. Programmeerimisplukkide allalaadimiste lehel tuleb valida “*HiTechnic Prototype Sensor Block for Mindstorms NXT Software*” ning seejärel on võimalik sisestada oma nimi ja e-mail (ei ole kohustuslik), et saada teavitusi uuenduste, uudiste ja uute näidisprogrammide kohta. Plokk on koos lisadega ZIP formaadis ning kasutamiseks tuleb see lahti pakkida.

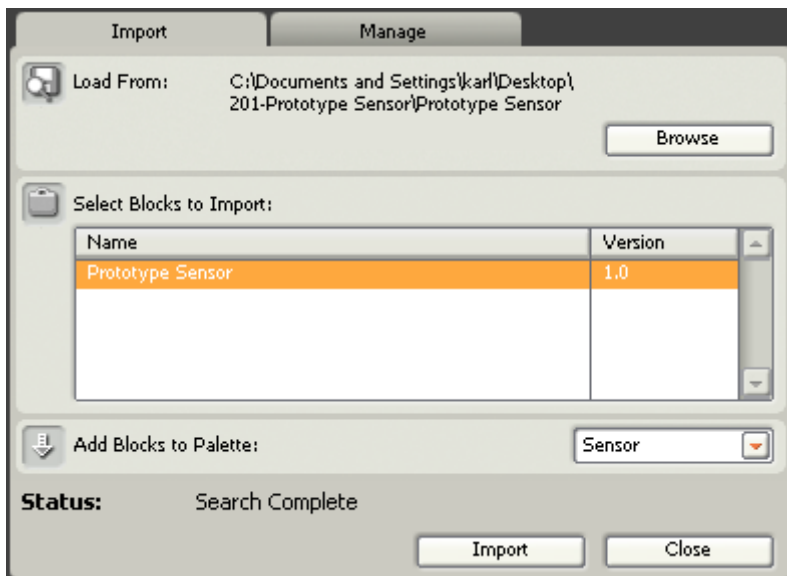
Ploki importimiseks tuleb kõigepealt käivitada NXT programm ning valida menüü ribast “*Tools*” ning avanenud valikutest vajutada „*Block Import and Export Wizard*“ peale (vt. Joonis 12).



**Joonis 12.** NXT prototüüpimislaua programmeerimisploki importimine

Avanenud uues aknas (vt. Joonis 13), vajutada nupule “*Browse*”, ning valida kaust, kuhu eelnevalt NXT prototüüpimislaua plokk lahti pakiti. Seejärel vajutada “*Prototype Sensor / 1.0*” peale, mis märgib selle aktiivseks (taust muutub oranžiks). “*Add Blocks to Palette*”st

valida võimalikkudest variantidest „*Sensor*“, mis ei ole küll kohustuslik, kuid seal oleks õige sensorite programmeerimisplokke hoida. Kui kõik vajalik on tehtud, siis muutub „*Import*“ nupp aktiivseks ning sinna vajutades imporditakse „*Prototype Sensor*“ plokk NXT programmi. Juhul kui see plokk on juba eelnevalt sisestatud, siis küsib programm kas tahate eelmise üle kirjutada või katkestada importimine.



**Joonis 13.** NXT prototüüpimislaua programmeerimisploki valimine

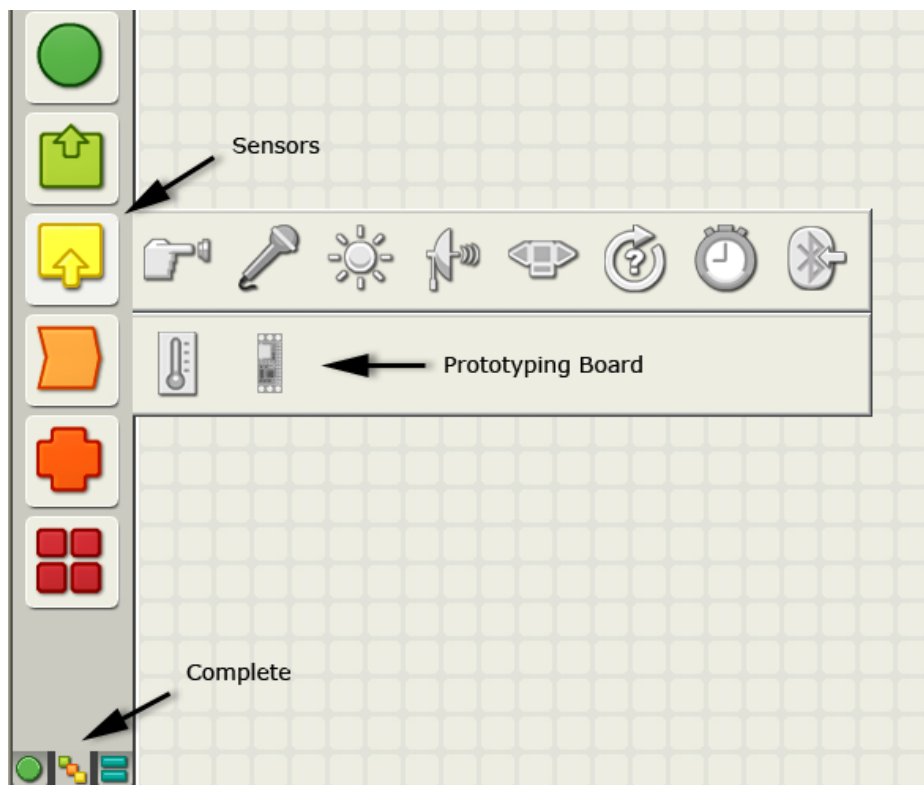
Eduka importimise kontrollimiseks võib vaadata, kas see plokk on tekkinud teiste sensorite kõrvale NXT graafilises keskkonnas. Õnnestumise korral võib hakata seda oma programmides kasutama.

Kuna NXT prototüüpimislaua digitaalsete ühenduste aadressid on arvu 2 astmed, siis on meil vaja ka plokki arvu astmesse tõstmiseks. Selle saab alla laadida aadressilt [23]. Tuvastamiseks, kas digitaalsisendisse saadetakse midagi, tuleb kasutada bitiloogika plokki, mille leiab aadressilt [24]. Plokkide lisamine käib täpselt samamoodi, nagu eelnevagi, kuid valikust „*Add Blocks to Palette*“ on soovitatav valida „*Data*“, mille all paiknevad ka teised matemaatilised tehted.

### 2.2.2 NXT-G prototüüpimislaua programmeerimisploki kasutamine

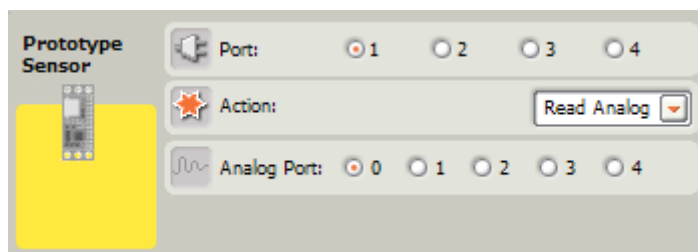
NXT prototüüpimislaua programmeerimisploki kasutamine võib esmapilgul tunduda suhteliselt keerukana, kuid kui juba mõned programmid valmis saada, siis edaspidi on see juba palju lihtsam. Seda on võimalik konfigureerida vedadades juhtmeid, mis sisaldavad tööks

vajalikke andmeid või kasutada selleks mõeldud seadistamisblokke. Ploki leiab (vt. Joonis 14) liikudes all vasakus servas asuvalle “*Complete*” menüüle, sealt edasi valida “*Sensors*” ning avanevast loetelust valida “*Prototyping Board*”. Järgnevalt on selgitatud analoog/digitaalsisendite ja digitaalväljundite konfigureerimisplokke.



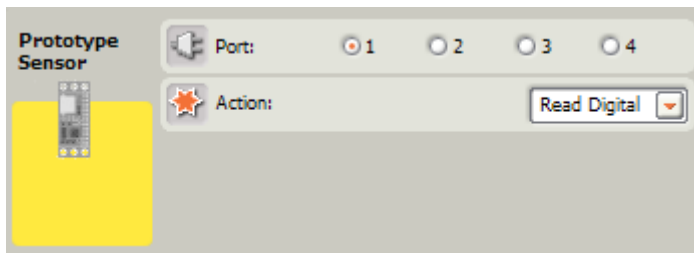
**Joonis 14.** NXT prototüüpimislaua programmeerimisploki valimine

Analoogsisendi seadistamiseks (vt. Joonis 15) tuleb valida “*Action*” rippmenüüst “*Read Analog*”. Valida “*Port*”, kuhu on ühendatud NXT prototüüpimislaud ning määrata “*Analog Port*”, mida soovitakse sisendiks lugeda.



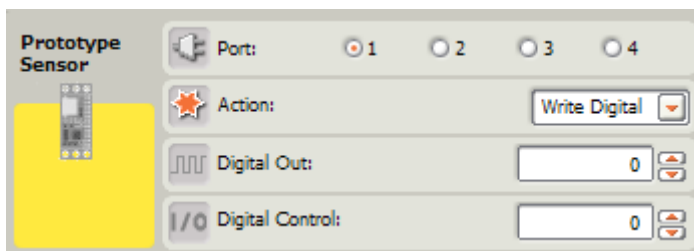
**Joonis 15.** Analoogsisendi programmeerimisblokk [10]

Digitaalsisendi seadistamiseks (vt. Joonis 16) tuleb valida “*Action*” rippmenüüst “*Read Digital*” ning valida “*Port*”, kuhu on ühendatud NXT prototüüpimislaud.



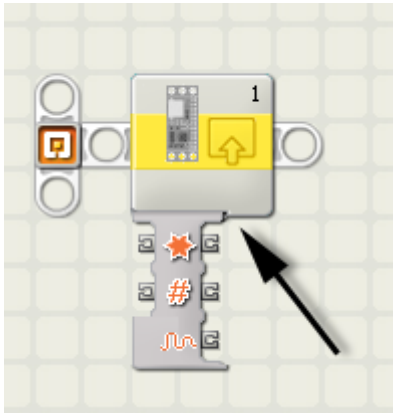
**Joonis 16.** Digitaalsisendi programmeerimisblokk [10]

Digitaalväljundi seadistamiseks (vt. Joonis 17) tuleb valida “Action” rippmenüüst “Write Digital” ning valida “Port”, kuhu on ühendatud NXT prototüüpimislaud. Seejärel tuleb määrata “Digital Out”, milles kirjeldatakse B0-B5, mis aktiveeritakse kui väljund ning “Digital Control”, mis määrab ära, millised neist on kui väljundid ja millised sisendid. Arvu, mida sinna kirjutada saab, tuleb arvutada Tabeli 2 järgi. Näiteks kui tahetakse konfigureerida nii, et B0, B3 ja B5 on väljundid, siis tuleb kokku liita arvud  $01 + 08 + 32$  ning saadud 41 kirjutada “Digital Control” väljale. Kahendsüsteemis on see arv 101001, ehk soovitud väljundid on märgitud numbriga 1, sisendid numbriga 0.



**Joonis 17.** Digitaalväljundi programmeerimisblokk [10]

Juhtmete kasutamiseks vajalikku rippmenüüd saab avada vajutades programmeerimisbloki alumisele äärele (vt. Joonis 18), mille tulemusena tekib lisaplokk sisendi- ja väljundi ühendamiseks (vt. Joonis 19). Neid on võimalik juhtmete abil kasutada oma programmi täiustamiseks. Ühenduste puhul määrab programm automaatselt juhtmed nelja eri värvi. Halli juhtme puhul võib järeldada, et midagi on valesti, ning ühendust ei saa kasutada (näiteks erinevad andmetüübid). Arvude edastamiseks kasutatakse kollast, loogikaväärtuste puhul rohelist ning teksti jaoks punast juhet.



**Joonis 18.** Programmeerimisploki algne seisund

Järgnevas nimekirjas on selgitatud ploki kasutusvõimalused (vt. Joonis 19):

1. Numbri ühendamine, kuhu porti on NXT prototüüpimislaud ühendatud.
2. Ploki tegevuse ühendamine (digitaal/analoogsisendi lugemine või digitaalväljundi kirjutamine).
3. Määrab millisest analoogsisendist loetakse (A0-A4).
4. Määratakse 6-bitine väärtus digitaalväljundiks.
5. Digitaalse sisendi/väljundi määramise kontroll.
6. Väljastatakse analoogsisendisse digitaalseks konverteeritud väärtus.
7. Väljastatakse digitaalsisendisse loetud väärtus.



**Joonis 19.** Programmeerimisblokk [10]

Olukordades, kus on vaja muuta tsükliliselt olenevalt digitaalväljundeid või lugeda erinevaid analoogsisendeid, on vajalik programmeerida kasutades juhtemeid. Juhtudel, kus NXT

prototüüpimislaualt loetakse digitaalsisendit või kirjutatakse digitaalväljundisse kogu aeg sama väärtust, võib kasutada selleks ettenähtud seadistamisblokke. Näiteks kui on vaja aktiveerida järjest digitaalühendused B0 - B5, oleks mõistlik kasutada programmeerimisel juhtmeid, millede kaudu saab ühendada omavahel tsükli toimumiskordade arvu ja digitaalväljundi seadistuse. Alternatiiv sellele oleks seadistusplokkide kasutamine, kuid sel juhul peaks olema kuus erinevalt seadistatud programmeerimisplokki järjest. Kui on olemas mõlemad seadistused ehk on nii juhtme, kui ka seadistamisblokiga, siis kasutatakse juhtmelt tulevaid andmeid.

## **2.3 NXT prototüüpimislaua programmeerimine keeles NXC**

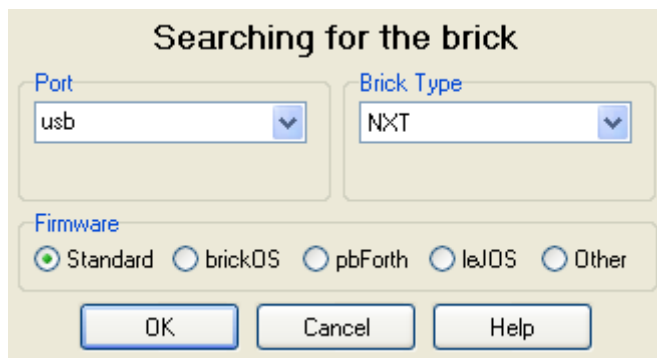
NXC on C-keelele sarnase süntaksiga programmeerimiskeel, mis baseerub Next Byte Codes assemblerkeelele [21]. Kuna NXC keel asub madalamal tasemel kui NXT-G keel, siis võimaldab see programmil joosta umbes kolm korda kiiremini [25], mis on prototüüpimislaua kasutuse juures suhteliselt tähtis. NXC keeles on lahendatud ka NXT-G keele üks suuremaid kitsaskohti - võimalus koostada kompleksseid programme. NXC keele põhiliste funktsioonide tutvustamine järgnevates peatükkidest eeldab, et lugejal on algteadmised tekstipõhises programmeerimises.

### **2.3.1 Programmeerimiskeskond Bricx Command Center**

Bricx Command Center (edaspidi lihtsalt BricxCC) on integreeritud arenduskeskkond, mis võimaldab koodi redigeerimist, selle paremat haldamist ning kompileerimist. BricxCC keskkonnas on võimalik programmeerida Not Quite C (NQC), Next Byte Codes (NBC) Not eXactly C (NXC), Pascal, C/C++ ning Java keeles. NBC jaoks on olemas ka koodi silumisvõimalus (ing. *debugging*). BricxCC keskkonna kaudu on võimalik valmistatud programm NXT põhiplokki alla laadida.

NXT põhiploki ühendamiseks BricxCC keskkonnaga valida ülemiselt menüüribalt “*Tools*” ning avanenud valikutest vajutada nupule “*Find Brick*”. Avanenud aknast (vt. Joonis 20) valida seadistuse “*Port*” variantidest “usb” ning “*Brick Type*” puhul “NXT”.





**Joonis 20.** Põhiploki leidmine Bricx Command Center keskkonnas

Eduka põhiploki leidmisel aktiveeruvad nupud, milledega saab kirjutatud programmi sinna alla laadida, käivitada ja peatada. Juhul kui programmi kompileerimisel tuleb mõni viga, siis kuvatakse veateade alumisele ribale.

### 2.3.2 NXC Standardteek

NXC standardteek koosneb funktsioonidest, konstantidest ja makrodest, mida on võimalik programmis kasutada. Standardteek on defineeritud eraldi spetsiaalses päisefailis, mis on lisatakse automaatselt programmi kompileerimisel. BricxCC arenduskeskkond oskab kasutada seda päisefaili koodi markeerimiseks ning automaatse sõnalõpetuse võimaluse jaoks. NXC standardteegi dokumentatsiooni leiab aadressilt [26], mis on genereeritud veebileheks kasutades *doxygen*'i. Sealt leiab kirjeldusi funktsioonide, konstantide ja makrode kohta ning ka näited nende kasutamisest. NXT prototüüpimislauda jaoks kasutatavad funktsioonid asuvad *LowSpeed* moodulis [27]. Järgnevalt on selgitatud NXC standardteegi tähtsamate funktsioonide kasutus NXT prototüüpimislauda seisukohalt.

### 2.3.3 Programmeerimine NXC's

NXC programm koosneb ühest põhilisest funktsioonist `main()`, mis käivitatakse NXT juhtimisbloki poolt ning see võib omakorda välja kutsuda erinevaid programmeerija poolt kirjeldatud funktsioone. Kui kasutada programmeerimiseks mingit muud keelt kui NXT-G, siis tuleb põhiplokile öelda, et pordis, kuhu on ühendatud NXT prototüüpimislauda, oleks aktiveeritud digitaalne ühendus I<sup>2</sup>C ja 9.0v. Selleks saab kasutada NXC funktsiooni `SetSensorLowSpeed`, kuhu antakse argumentideks port (määratletud konstantidena S1-S4) ja tõeväärtus, kas I<sup>2</sup>C ja 9.0v aktiveeritakse või mitte, mis on vaikimisi tõene. Seega kui NXT prototüüpimislauda asub põhiploki pordis 1, siis peame alustama oma `main()` funktsiooni järgmiselt:

```
task main() {  
    SetSensorLowspeed(S1);  
}
```

Põhiliseks funktsiooniks, mida saab kasutada prototüüpimislauga suhtlemiseks, on I2CBytes. See algatab operatsiooni, mis võimaldab kirjutada ja lugeda andmeid. I2CBytes on kõrgema taseme meetod, mis koondab enda alla põhilised I2C tüüpi funktsioonid [28]. Kontrollimiseks, kas NXT prototüüpimislaud on valmis vastu võtma järgmist ülesannet, saab teha funktsiooniga I2CCheckStatus [29]. Järgnevalt tuleb nende funktsioonide kasutusõpetus.

Funktsiooni I2CBytes kirjutamist ja lugemist juhib massiiv, mis tuleb eelnevalt luua ning täita sobivate andmetega. Selle tegemiseks võib kasutada NXC standardteegist pärinevat funktsiooni ArrayInit [30], parameetritega milleks on massiivi nimi, kõikide elementide algne väärtus ja massiivi pikkus. Näiteks käsuga ArrayInit(kirjuta, 0, 3) luuakse massiiv “kirjuta”, millel on kolm elementi ja nad kõik on väärtusega null. Järgnevalt tuleb massiivis paika panna NXT prototüüpimislaua aadress. Selleks tuleb programmi kirjutada käsk kirjuta[0] = 0x02. Tahtmisel kirjutada registrisse, mis hoiab endas digitaalühenduste sisend/väljund kontrolli (vt. Tabel 1), peab massiivi teine väärtus olema kirjeldatud kui kirjuta[1] = 0x4E. Kolme esimese digitaalühenduse väljundiks määramiseks peab viimane element massiivis olema defineeritud kui 0x07, seega kirjuta[2] = 0x07. Registrist lugemiseks tuleb ära jätta massiivis asetsev viimane element. I2CBytes funktsiooni väljakutsumise parameetriteks on pordi number, kuhu NXT prototüüpimislaud on ühendatud, massiiv, mis juhib kirjutamist/lugemist, baitide arv lugemisel ja eeldefineeritud masiiv, kuhu paigutatakse loetud andmed. Funktsioon tagastab tõeväärtuse transaktsiooni edukuse kohta.

Funktsiooni I2CCheckStatus saab kutsuda välja argumendiga, milleks on pordi number, kuhu NXT prototüüpimislaud on ühendatud. Juhul kui funktsioon tagastab arvu 0, ehk koodi NO\_ERR’ile, siis võib edasi tegutseda. Teine võimalus on, et see tagastab arvu 32, ehk numbrilise väärtuse STAT\_COMM\_PENDING’ule, mille puhul tuleb oodata natukene, ning uuesti proovida.

Mõistmaks eelnevaid selgitusi paremini, tuleb järgnevalt näide NXC programmi kohta, mis aktiveerib digitaalühenduse B0. Kui sinna on näiteks ühendatud LED tuli, siis see läheb põlema. Kommentaarid on märgistatud kaldkriipsudega.

```

//defineerime massiivid, mis hakkavad hoidma endas kirjutamist juhtivat
//tegevust ja loetud andmeid
byte kirjuta[];
byte loe[];
task main() {
    //Aktiveerime põhiplokil I2C ja 9.0v
    SetSensorLowspeed(S1);
    //teeme "kirjuta" massiivi 3 elementi ja väärtustame nullidega.
    ArrayInit(kirjuta, 0, 3);
    //väärtustame massiivi prototüüpimislaua aadressiga, digitaalühenduste
    //kontrolli registri ja väärtusega mis sinna registrisse kirjutatakse.
    kirjuta[0] = 0x02;
    kirjuta[1] = 0x4E;
    kirjuta[2] = 1;
    //kuna tagastavate baitide suurus on null siis:
    int baite = 0;
    //Kirjutame prototüüpimislaua registrisse, et B0 on digitaal väljund
    I2CBytes(S1, kirjuta, baite, loe);
    //Ootame, kuni kirjutamine valmis on
    while(I2CCheckStatus(S1) != 0){
        //Kuna NXT prototüüpimislaud uuendab seisuga 4ms tagant, siis peaks
        //ka selle aja ootama enne järgmist käsku. Programmi jooksmine ja läbi
        //protokolli suhtlus võtab ka aega, seega 3ms oodata on piisav.
        Wait(3);
    }
    //Vahetame digitaalühenduste kontrolli registri digitaalväljundite
    //registri vastu ning kuna sooviks on aktiveerida B0, siis sobib ka sinna
    //kirjutada väärtus 1.
    kirjuta[1] = 0x4D;
    //Kirjutame väärtuse 1 digitaalsete väljundite alla
    I2CBytes(S1, kirjuta, baite, loe);
    //Registrisse jäävad andmed nii kauaks alles, kuni prototüüpimislauas
    //voolu on, seega jääb ka LED tuli põlema ja B0 aktiveerituks. Meil mõtet
    //programmi tööle jätta, vaid võime lasta sellel lõppeda.
}

```

Järgmises punktis on võimalik eelnevate peatükkide põhjal lahendada praktilisi ülesandeid, mis on jagatud erinevatesse raskusastmetesse.

### 3. Ülesanded

Käesolevas peatükis on koostatud ülesanded, mida on võimalik lahendada kasutades NXT prototüüpimislauda, maketiplaani, LED-tulesid ning mõningaid juhtmeid, takisteid. Ülesanded 1-3 on erineva raskusastmega, alustades kergeimast ja lõpetades raskeimaga. Algtaseme ülesande juures on lähtutud kõige lihtsamatest toimingutest, mida NXT prototüüpimislauga teha võiks. See nõuab minimaalset programmeerimist ning väliste elementide kasutust. Järgmised tasemed nõuavad juba rohkem programmeerimist ja elektrooniliste komponentide kasutust.

Ülesanded on kirja pandud kindla struktuuri järgi. See lihtsustab erinevatest bakalaureusetöödest ülesannete kokkupanemist ja kasutamist õppematerjalina, iga ülesanne koosneb järgmistest osadest:

- Tase – näitab millise keerukusega ülesandega on tegu (lihtne, keskmine, raske).
- Eesmärk – kirjeldab, mida antud ülesande lahenduse käigus õpitakse.
- Vajalikud vahendid ja teadmised – kirjeldab, milliseid vahendeid ja eelteadmisi on ülesande lahendamiseks vaja.
- Ülesande püstitus – määrab ära ülesande olemuse ning kirjeldab, mida oodatakse lahenduseks.
- Lahenduse idee – annab juhiseid, kuidas konkreetset ülesannet tuleks lahendada.
- Üks võimalik lahenduse variant NXT-G keeles – esitatud üks ülesande võimalikest lahendusavariantidest. Ülesanded on lahendatud LEGO MINDSTORMS Education NXT v2.0 keskkonnas ja ülesannate lahendusfailid asuvad käesoleva bakalaureusetöö lisas oleva CD plaadi peal (vt. Lisa 1).
- Üks võimalik lahenduse käik NXC keeles - tähtsamatest kohtadest tehtud kokkuvõtte selle ülesande lahendamiseks. Ülesannate täielikud lahendusfailid koos kommentaaridega asuvad käesoleva bakalaureusetöö lisas oleva CD plaadi peal (vt. Lisa 1).
- Vooluring - Kuna kõik ülesanded koosnevad kahest osast - käivitatavast programmist ja vooluringist siis on ka sellest olemas seletus, joonis ja skeem, kuidas peaksid maketiplaanil elemendid ühendatud olema.
- Märkused – kirjeldab erinevaid probleeme, mis ülesande lahendamise käigus võivad esile kerkida, ning annab lahendusi nende vältimiseks.

- Ideed ülesande muutmiseks - kirjeldatakse, mida muuta ülesande juures, et selle keerukust tõsta. Pakutakse ideid, mida võiksid õpilased lisaks proovida, kui neil esialgne ülesanne juba lahendatud on.

### **3.1 Ülesanne 1 - Vilkuv LED**

#### **Tase:**

Lihtne, sobib NXT prototüüpimislauaga esmaseks tutvumiseks.

#### **Eesmärk:**

Tutvuda NXT prototüüpimislaua, maketiplaani ja vooluahelaga. Teha esimesed sammud digitaalväljundiga.

#### **Ülesande täitmiseks vajalik:**

- LEGO MINDSTORMS NXT juhtplokk
- HiTechnic NXT Prototüüpimislaud
- Kaabel NXT prototüüpimislaua ja juhtploki ühendamiseks
- NXT-G programmeerimiskeskond, millesse on imporditud NXT prototüüpimislaua plokk või Bricx Command Center
- 2 juhet
- 1 LED tuli (vajadusel ka takisti)

#### **Ülesande püstitus:**

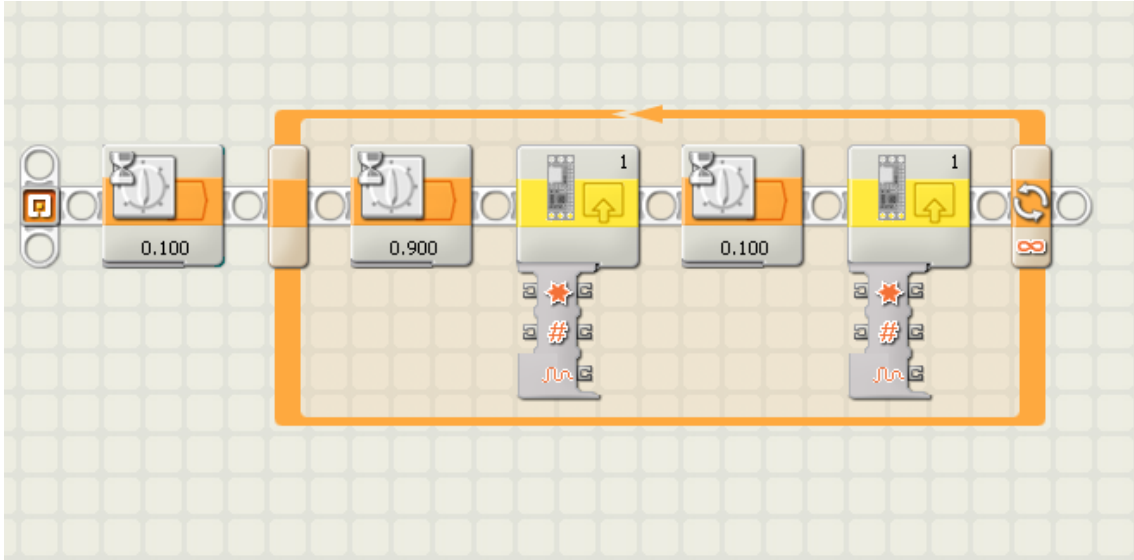
Panna LED tuli iga sekundi tagant 0.1 sekundiks põlema, kasutades NXT prototüüpimislauda ja LEGO MINDSTORMS NXT juhtploki.

#### **Lahendusidee:**

Teha juhtmete ja maketiplaani abil vooluring, kus LED tule anood on ühenduses digitaalliidese B0-ga ning LED'i katood ühendatud NXT prototüüpimislaua GND ühendusega. Panna tuli vilkuma programmiga, mis kasutab tsüklit, ootamisfunktsiooni ja digitaalväljundit.

### Üks võimalik lahendusvariant NXT-G keeles:

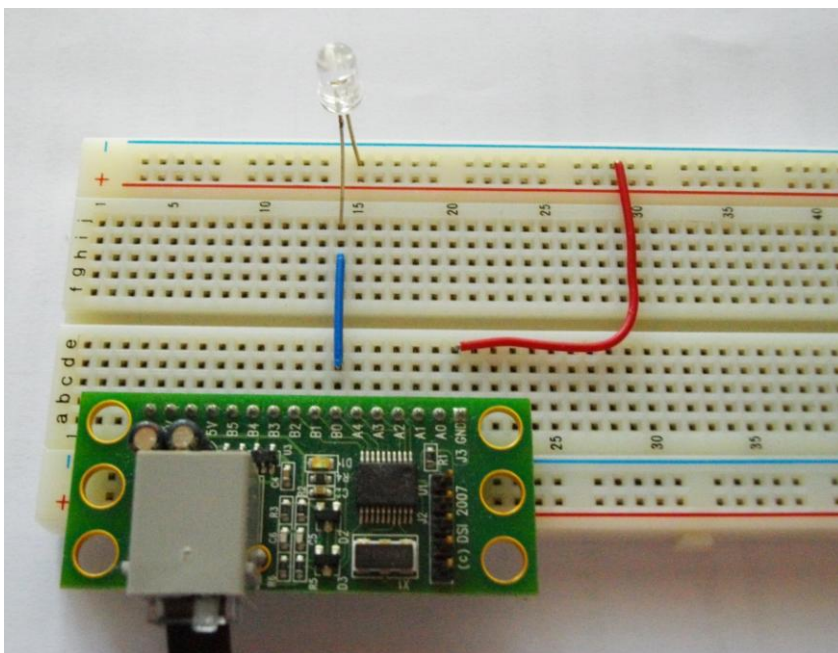
Lahenduse programmi osa peaks välja nägema sarnane järgneval joonisel (vt. Joonis 21) oleva programmiga. Lahendusfail: yll.rbt



**Joonis 21.** Ülesande 1 lahendus NXT-G keskkonnas

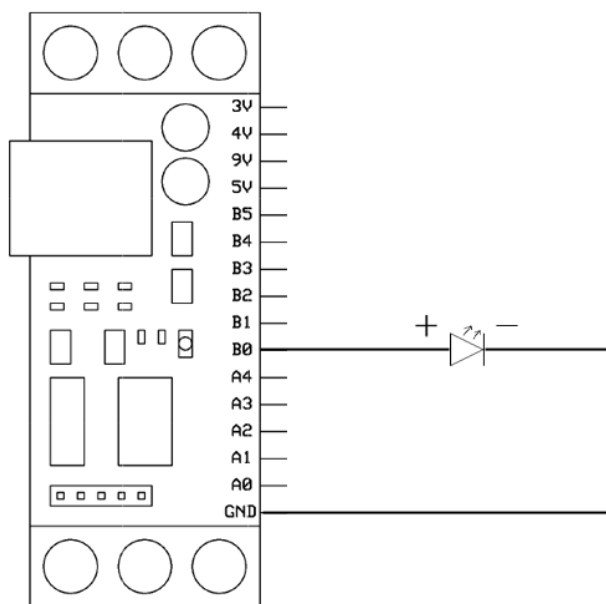
### Vooluring:

Maketiplaanil paiknev vooluring peaks välja nägema sarnane järgneval joonisel (vt. Joonis 22) olevaga.



**Joonis 22.** Ülesande 1 lahendus maketiplaanil

Selle vooluringi skeem on kujutatud joonisel nr 23. Kujutatud on ka LED tule anood „+” ja katood „-”.



**Joonis 23.** Ülesande 1 lahenduse skeem

#### Üks võimalik lahenduskäik NXC keeles:

Kõigepealt kasutada funktsiooni SetSensorLowSpeed I<sup>2</sup>C ja 9.0v aktiveerimiseks. Seejärel initsialiseerida massiiv ArrayInit'iga ning täita see väärtustega 0x02, 0x4E, 0x01 (NXT prototüüpimislaua aadress, digitaal sisend/väljundi seadistuse registri valimine, B0 väljundiks määramine). Käivitada I2CBytes funktsiooni ja loodud massiivi kasutades NXT prototüüpimislauda kirjutamine. Luua uus massiiv sarnaselt eelmisega, ning täita see väärtustega 0x02, 0x4D (NXT prototüüpimislaua aadress, digitaalväljundi registri valimine). Teha lõpmatu tsüklitel, mis algab uue massiivi viimase elemendi 0x01 väärtustamisega ning kus kutsutakse välja I2CBytes välja koos selle massiiviga. Seejärel oodatakse 0.1 sekundit kasutades Wait funktsiooni [31], omistatakse massiivis viimasel kohal asuv elemendi väärtuseks 0x00, kutsutakse uuesti I2CBytes välja ning seejärel oodatakse 0.9 sekundit. Lahendusfail: y11.nxc

#### Märkused:

- Enne käivitamist kontrollida, kas LED tuli on õigesti ühendatud, ehk kas anood on ühendatud B0'iga ja katood GND'ga. Vastasel juhul põleb tuli läbi.

**Ideed ülesande muutmiseks:**

Muuta programmi nii, et iga viienda vilkumise järel käib NXT põhiplokis piiks.

## 3.2 Ülesanne 2 - Jooksvad tuled

**Tase:**

Keskmine. Sobib neile, kes on juba natuke tutvunud NXT prototüüpimislauaga.

**Eesmärk:**

Teha selgeks NXT prototüüpimislaua programmeerimisbloki kasutamine juhtmetest tulenevate sisendite ja väljundite abil. Digitaalühenduste aadressiloogika ja kahendsüsteemi selgitamine praktikas.

**Ülesande täitmiseks vajalik:**

- LEGO MINDSTORMS NXT juhtplokk
- HiTechnic NXT Prototüüpimislaut
- Kaabel NXT prototüüpimislaua ja juhtploki ühendamiseks
- NXT-G programmeerimiskeskond, millesse on imporditud NXT prototüüpimislaua plokk või Brixx Command Center
- 1 juhet
- 6 punast LED tuld
- 6 takistit  $68\Omega$ , 1/8W

**Ülesande püstitus:**

Panna digitaalliidestesse ühendatud LED tuled jooksmas edasi-tagasi nagu need jooksid seriaalis "Knight Rider" autol KITT (vt. video [32]). Ülesande aluseks on võetud allikas [33].

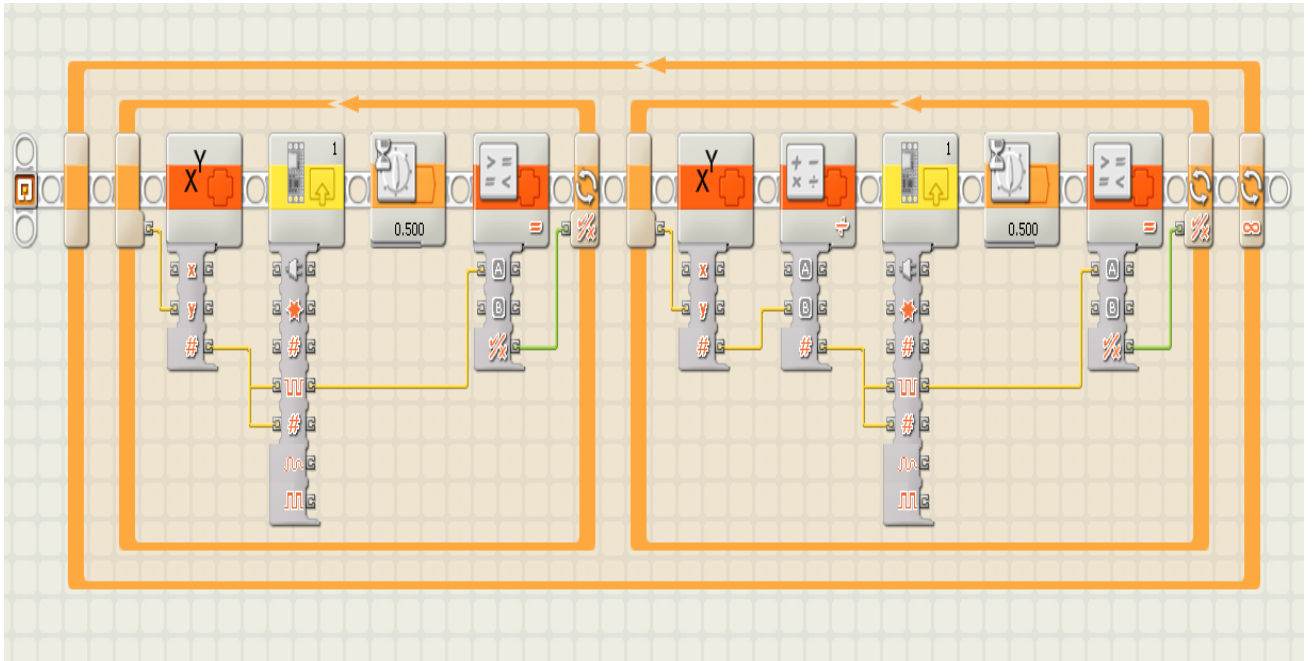
**Lahendusidee:**

Ühendada LED tuled digitaalühenduste B0 - B5 ning nende katoodeid NXT prototüüpimislaua GND ühendusega. Kasutada ühendamisel  $68\Omega$  ja vähemalt 1/8W takisteid, kuna punased LED tuled töötavad pingetel 1.63v - 2.03v [34], kuid NXT prototüüpimislaua digitaalväljundist tuleb teatavasti välja 3.3v. Koostada programm mis aktiveerib järjest digitaalühendusi ning viimaseni jõudes hakatakse aktiveerimisega nõ tagasi tulema.



### Üks võimalik lahendusvariant NXT-G keeles:

Lahenduse programmi osa peaks välja nägema sarnane järgneval joonisel (vt. Joonis 24) oleva programmiga. Lahendusfail: yl2.rbt



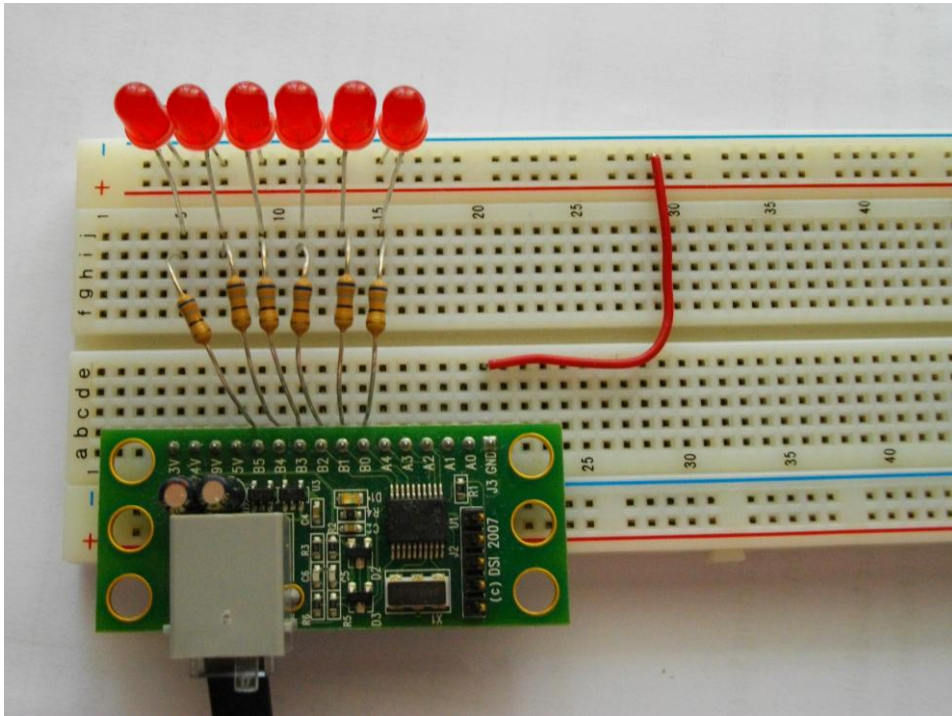
**Joonis 24.** Ülesande 2 lahendus NXT-G keskkonnas

### Üks võimalik lahenduskäik NXC keeles:

Alustada sarnaselt eelmise ülesande lahenduskäiguga, kuid digitaal sisend/väljundi registrisse kirjutada väärtus 0x3F, ehk B0 - B5 on määratud väljunditeks. Teha tsükkel, mis aktiveerib järjest digitaalväljundeid, kirjutades sinna registrisse erinevaid väärtusi. Digitaalühendusi saab ühevõrra suurendada korrutades selle eelnevat väärtust kahega, ehk 0x02. Kontrollida, kas arv on juba jõudnud 0x20 (B5 aktiveerimiskäsk) ning seejärel hakata tagasi tulema kasutades jagamistehet ning 0x01 korral alustada jälle korrutustega. Iga tsükli lõpus tuleb oodata 0.1 sekundit. Lahendusfail: yl2.nxc

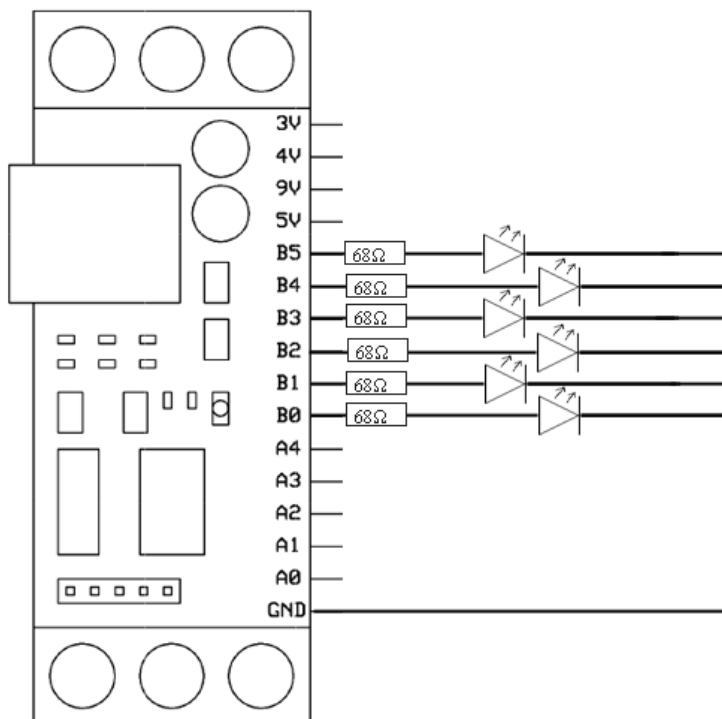
### Vooluring:

Maketiplaanil paiknev vooluring peaks välja nägema sarnane järgneval joonisel (vt. Joonis 25) olevaga. Selles vooluringis on kasutatud 1/4W võimsusega takisteid, kuid sobilikud on ka 1/8W takistid.



**Joonis 25.** Ülesande 2 lahendus maketiplaanil

Selle vooluringi skeem on kujutatud joonisel nr 26.



**Joonis 26.** Ülesande 2 laheduse skeem

**Märkused:**

- Kui takistid panemata jätta, siis punase LED'i eluaeg lüheneb või see põleb läbi. Juhul kui kasutada mõnda teist värvi LED tuld, siis vaadata aadressilt [34] nende pingeid ning arvutada sobiv takistus valemiga  $R = (V_P - V_L)/I$  [35], kus  $V_P$  on prototüüpimislauast tulev pinge ning  $V_L$  LED tulega sobiv pinge.

**Ideed ülesande muutmiseks:**

Muuta programmi nii, et tuled edasi-tagasi jooksmise asemel alustavad põlemisega mõlemast äärest ning saavad keskel kokku. Seejärel lähevad lahku tagasi.

### 3.3 Ülesanne 3 - Valguse andur

**Tase:**

Raske. Sobib neile, kes on põhjalikumalt tutvunud NXT prototüüpimislauaga ja omavad vastavat kogemust programmeerimisel ja takistite kasutamisel.

**Eesmärk:**

Tutvustada NXT prototüüpimislaua analoogsisendi kasutamist ning selgitada fototakisti tööpõhimõtet.

**Ülesande täitmiseks vajalik:**

- LEGO MINDSTORMS NXT juhtplokk
- HiTechnic NXT Prototüüpimislaud
- Kaabel NXT prototüüpimislaua ja juhtploki ühendamiseks
- NXT-G programmeerimiskeskond, millesse on imporditud NXT prototüüpimislaua plokk või Bricx Command Center
- 5 juhet
- 2 LED tuld (vajadusel ka takistid)
- 1 10000Ω takisti
- 1 fototakisti

## Ülesande püstitus:

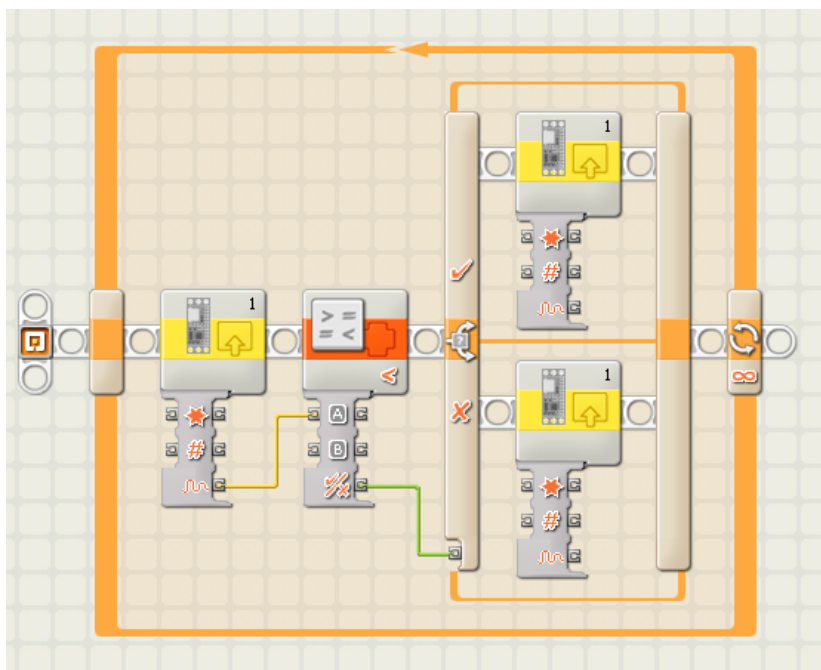
Oletame, et robot sõidab ringi kasutades NXTCam'i enda teejuhina. Kuna hämaruses tunnetab see kaamera värve erinevalt [36], siis on vaja selles suhtes midagi ette võtta. Ehitada andur, mis kontrollib, kas on läinud liiga hämaraks, juhul kui on, siis lülitada sisse 2 LED tuld.

### Lahendusidee:

Ühendada 2 LED tuld NXT prototüüpimislauga sarnaselt eelnevate ülesannetega, kuid seekord võiks kasutada ainult ühte digitaalühendust, kuna tuled põlevad alati koos. Seejärel ühendada toiteallikas, mille pingeks on 3.3v, ja fototakisti analoogsisendiga. Seejärel ühendada analoogsisend kasutades 10kΩ takistit GND'ga. Fototakisti muudab oma takistuse suurus vastavalt sellele, kui palju talle hetkel valgust peale paistab. Täiesti pimedas on see vastava fototakisti maksimaalne ning valges minimaalne. Teha programm, mis kontrollib, kas toiteallikast tulev pinge on muutunud väiksemaks fototakisti takistuse suurenemise tulemusena. Eelnevalt tuleb testida fototakistit, et teada saada milliseks muutub pinge täiesti valges ja pimedas.

### Üks võimalik lahendusvariant NXT-G keeles:

Lahenduse programmi osa peaks välja nägema sarnane järgneval joonisel (vt. Joonis 27) oleva programmiga. Lahendusfail: yl3.rbt



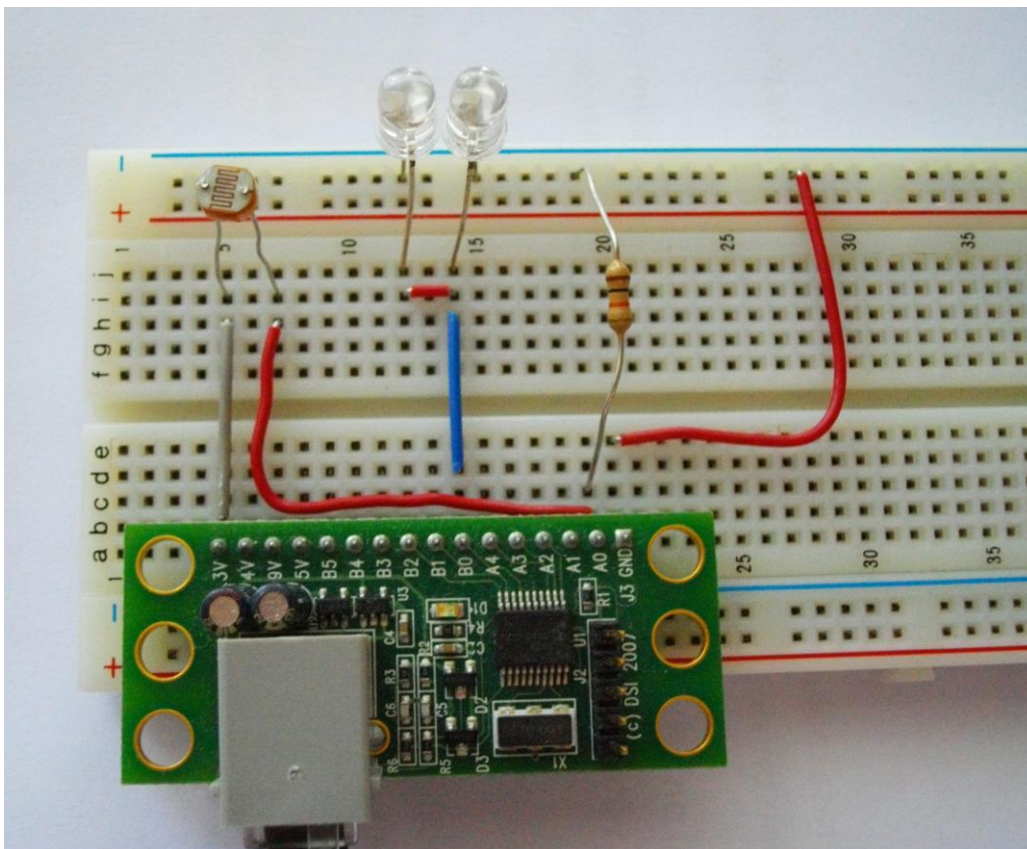
**Joonis 27.** Ülesande 3 lahendus NXT-G keskkonnas

### Üks võimalik lahenduskäik NXC keeles:

Alustada programmi sarnaselt eelmistele ülesannetele, kuid nüüd kasutada I2CBytes funktsiooni lugemiseks. Teha sarnane massiiv, kuid seal, kus pandi paika kirjutamiseks register, panna väljaks 0x42, ehk A0 ülemised 8 bitti. Sinna pöördumisel tagastatakse ka väljal 0x43 asetsevad A0 alumised 2 bitti, seega peab I2CBytesi funktsioonis eksisteerima arv 2 tagastatavate baitide suuruse parameetrina. Lõpmatus tsüklis teha koguaeg A0 mäluväljale päringuid ning kontrollida, kas tagastatud arv on võrreldes kalibreeritud suurusega liiga väike. Juhul kui on, panna I2CBytes'iga digitaalsisendites B0 ja B1 asuvad LED tuled põlema. Kui on läinud jälle valgeks, kustutada tuled. Lahendusfail: yl3.nxc

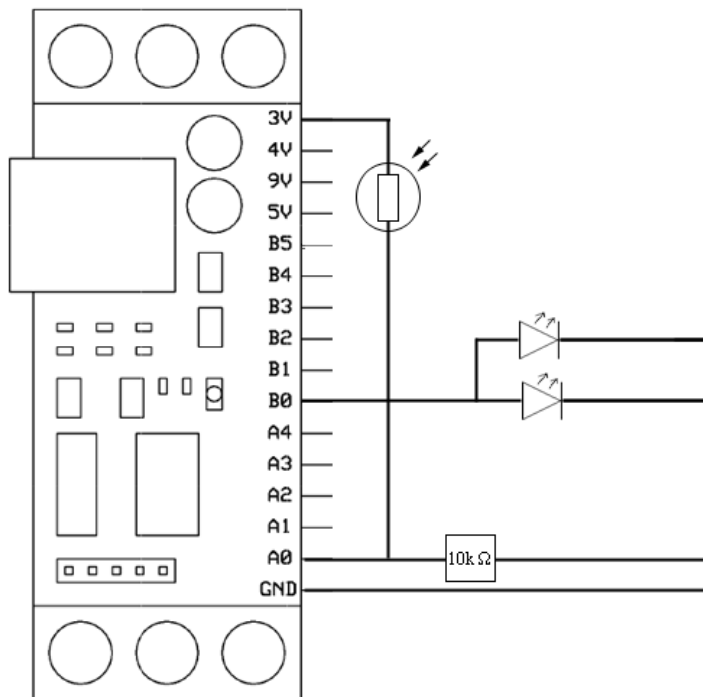
### Vooluring:

Maketiplaanil paiknev vooluring peaks välja nägema sarnane järgneval joonisel (vt. Joonis 28) olevaga.



**Joonis 28.** Ülesande 3 lahendus maketiplaanil

Selle vooluringi skeem on kujutatud joonisel nr 29.



**Joonis 29.** Ülesande 3 lahenduse skeem

**Märkused:**

- Töös kasutatav fototakisti varieerub takistuste vahemikus 3 - 20kΩ (tootekood MKY-54C348), kuid seda ülesannet on võimalik lahendada ka teistsuguse fototakistiga.
- Jadamisi LED tulede ühendamisel tuleks kasutada sama pingega LED'e. Vastasel juhul võib väiksema pingega töötav LED tuli läbi põleda.
- A0 töötab nagu voltmeeter, mis mõõdab punkti ja maa vahelist pinget. Sellepärast on vaja sinna panna ka 10kΩ takisti, et saaksime pingelangu. Näiteks kui valgust tuleb rohkem, siis fototakisti takistus väheneb, vool suureneb, kuid pingelang 10kΩ takisti peal suureneb.

**Ideed ülesande muutmiseks:**

Muuta programmi nii, et ainult teatud hämarus astme vahemikus lülitatakse sisse 2 LED tuld, kui see ületatakse, siis lülitada sisse 4 LED tuld.

### 3.4 Ülesanne 4 - Reaktsiooni aja mõõtmine

#### Tase:

Raske. Sobib neile, kes on põhjalikumalt tutvunud NXT prototüüpimislauaga ja omavad vastavat kogemust programmeerimises.

#### Eesmärk:

Tutvustada NXT prototüüpimislaua digitaalsisendi kasutamist ning lülite tööpõhimõtet.

#### Ülesande täitmiseks vajalik:

- LEGO MINDSTORMS NXT juhtplokk
- HiTechnic NXT Prototüüpimislaud
- Kaabel NXT prototüüpimislaua ja juhtploki ühendamiseks
- NXT-G programmeerimiskeskond, millesse on imporditud NXT prototüüpimislaua plokk või Bricx Command Center
- 4 juhet
- 1 LED tuld (vajadusel ka takistid)
- 1 Lüliti
- 1 10000Ω takistit

#### Ülesande püstitus:

Luu seade, mis suvalisel ajahetkel paneb LED tule põlema ning seejärel ootab kasutajapoolset lülile vajutust. Põlema panemiseks kuluv maksimaalne aeg on 5 sekundit. Aeg, mis kulub tule põlemapanekust kuni lülile vajutuseks kuvada ekraanile. Seejärel oodata 2 sekundit ning korrata reaktsiooni aja mõõtmist. Idee saadud materjalist [37].

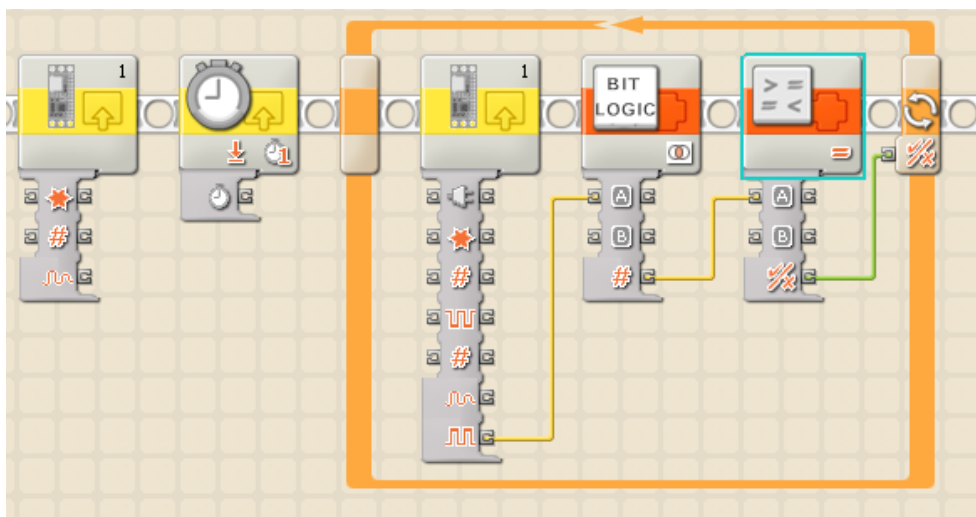
#### Lahendusidee:

Ühendada LED tuled digitaalühenduse B0 ning lüliti B1'ga. Teha programm, mis kasutab Random funktsiooni/blokki suvalise arvu saamiseks. Seejärel käivitada taimer ning võrrelda taimeri aega suvalise arvuga. Kui taimeri aeg ületab arvu, panna B0'is asuv LED-tuli põlema. Panna käima järgmine taimer, ning tsüklis kontrollida kas lülile vajutatakse. Juhul kui vajutatakse, siis lõpetada taimeri töö ning kuvada selle aeg ekraanil. Kustutada LED tuli ning oodata 2 sekundit järgmise reaktsiooni aja mõõtmise alguseni.



### Üks võimalik lahendusvariant NXT-G keeles:

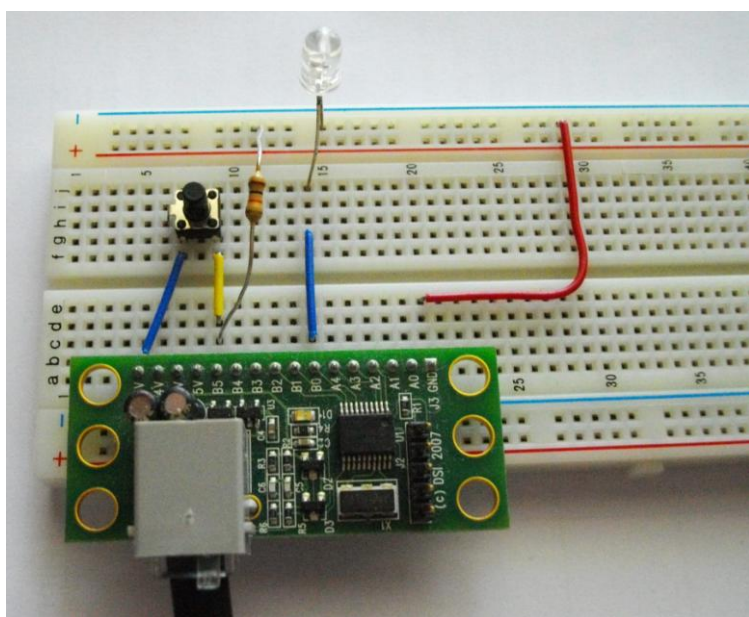
Lahenduse programmi osa, kus pannakse LED-tuli põlema ning kontrollitakse, kas nupule on vajutatud, peaks välja nägema sarnane järgneval joonisel (vt. Joonis 30) olevaga. Programmi ülejäänud osade valmistamisel ei tohiks probleeme tekkida, kui eelnevalt on lahendatud ülesanded 1, 2 ja 3. Lahendusfail: yl4.rbt



**Joonis 30.** Ülesande 4 lahendus NXT-G keskkonnas

### Vooluring:

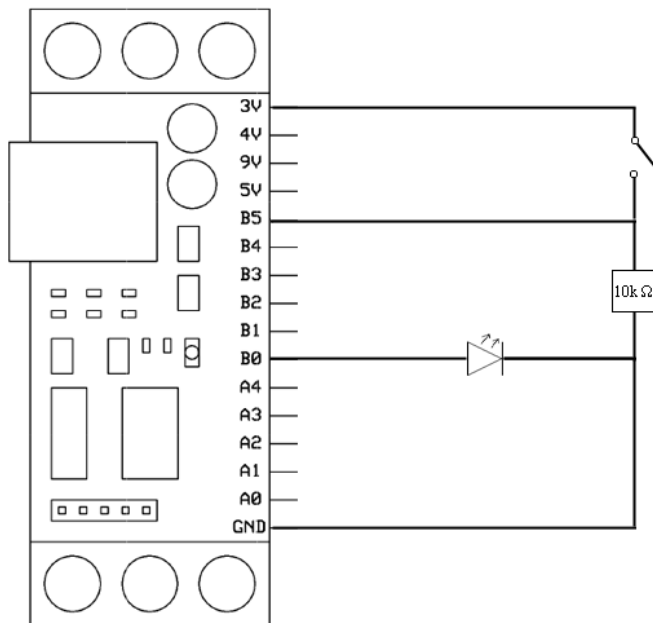
Maketiplaanil paiknev vooluring peaks välja nägema sarnane järgneval joonisel (vt. Joonis 31) olevaga.



**Joonis 31.** Ülesande 4 lahendus maketiplaanil



Selle vooluringi skeem on kujutatud joonisel nr 32.



**Joonis 32.** Ülesande 4 lahenduse skeem

### Üks võimalik lahenduskäik NXC keeles:

Sarnaselt eelmistele ülesannetele aktiveerida I<sup>2</sup>C ja 9.0v NXT prototüüpimislaua pordis. Kasutada NXC standardteegist funktsiooni Random [38] suvalise arvu saamiseks ning meetodit CurrentTick [39] hetke süsteemiaja teadasaamiseks. Oodata Wait funktsiooniga suvalisel teel saadud aeg ning fikseerida aeg. Teha päringuid digitaalsisendite väljale 0x4C kasutades funktsiooni I2CBytes ning bitiloogika tehtega kontrollida tagastatud arvu. Kuna lüliti asub digitaalühenduses B5, siis tehe, mis nupule vajutamist kontrollib on järgmine: if (saadudandmed & 32 == 32). Selle tõseks muutumisel teha uus päring hetke aja kohta ning lahutada sellest eelpool fikseeritud aeg. Kuvada vastus ekraanil kasutades TextOut [40] funktsiooni. Lahendusfail: yl4.nxc

### Märkused:

- Kasutada lüliti ja GND vahel kindlasti takistit, et hoida selles väärtust 0, ajal mil nupule pole vajutatud. Samuti aitab see kaasa lühisest hoidumiseks.

- Lahendada ülesannet ilma bitiloogika tehteid kasutamata, peab ühendama GND'ga ka kõik ülejäänud digitaalühendused, kuna vastasel juhul hakkavad nad mõõtmist segama.
- Töös on kasutatud 4-jalgset lülitit. Selle vajutamisel on omavahel ühendatud kõik jalad. Ülesande lahendamisel võib kasutada ka 2-jalgset lülitit.

**Ideed ülesande muutmiseks:**

Teha reaktsiooni aja mõõtmissüsteem selliseks, et seda ei oleks võimalik petta. Hetkel antakse reaktsiooni aeg 0,01 kui nuppu kogu aeg all hoida. Muuta programmi nii, et ta kontrollib digitaalsisendi muutust, mitte selle aktiveeritud olekus olemist. Samuti karistada 0,2 sekundiga neid, kes plõksutavad nupuga enne LED tule põlema minemist.

## Kokkuvõte

Käesoleva bakalaureusetöö eesmärgiks oli eestikeelse dokumentatsiooni loomine LEGO MINDSTORMS NXT'ga ühilduva prototüüpimislaua firmalt HiTechnic kohta. Töö andis põhjaliku ülevaate NXT prototüüpimislauast, selle tehnilistest aspektidest ning ka rakendusvõimalustest. Seda tööd on võimalik alusena kasutada tulevikus koolides erinevate tundide huvitavamaks muutmisel.

Bakalaureusetöö käigus valminud materjal läheb kasutusse Kooliroboti projekti raames, mistõttu on see koostatud võimalikult kergesti mõistetavalt. Sellest peaksid lihtsalt aru saama ka kesk- ja põhikooli õpetajad, õpilased. Töös on jälgitud teiste LEGO MINDSTORMS NXT'ga seonduvate tööde struktuuri ning on jaotatud kolmeks peatükiks.

Esimeses peatükk andis ülevaate prototüüpimise ning NXT prototüüpimislaua olemusest ning selle tähtsamatest komponentidest. Selgitasin paralleelselt lahti ka töös uuritava prototüüpimislaua tehnilised omadused ning selle suhtluse NXT põhiplokiga.

Teises peatükis on keskendusin NXT prototüüpimislaua kasutamise selgitamisele - nii mehaanilise kui ka programmeerimise osas. Viimases neist on lahti seletatud NXT prototüüpimislaua programmeerimine NXT-G ning natuke madalatasemelises Not Exactly C keeles.

Viimane peatükiks sisaldab endas nelja kolme erineva raskustasemega ülesandeid, mida on võimalik lahendada eelnevalt omandatud materjali põhjal. Ülesannete koostamisel on arvestatud, et need hõlmaksid NXT prototüüpimislaua põhilisi funktsioone. Esimesed kaks keskenduvad NXT prototüüpimislaua digitaalühenduste ja kahendsüsteemi loogika õpetamisele, kuid järgmised juba reaalse andurite ehitamisele. Ülesanded on kirjas kujul, mis lihtsustavad nende õppematerjalidena kasutamist.

# **HiTechnic NXT Prototype Sensor Board for LEGO MINDSTORMS NXT – overview and exercises**

Bachelor Thesis

Karl Kilgi

## **Abstract**

The aim of this bachelor thesis was to describe HiTechnic NXT Prototype Sensor Board and make practical tasks for its usage. This thesis has three main parts - an introduction to prototyping and overview of NXT Prototype Sensor Board, installation and programming guide for NXT-G and NXC, tasks for using the prototyping board.

In the first part the main focus is on the explanation of prototyping in electronics, main techniques of that and interacting it with NXT Prototype Sensor Board and its technical aspects.

In the second part can be found all that there is to be known about safe usage of prototype board and how to connect and program the prototype board in LEGO MINDSTORMS Education v2.0 environment and in Bricx Command Center.

The last part of the thesis contains four different tasks in order to give a practical approach for using and understanding the NXT Prototype Sensor board. The tasks are presented along with descriptions and images how to solve them.

This thesis is going to be a part of a teaching material that consists of all LEGO MINDSTORMS NXT related bachelor thesis. The purpose of the teaching material is to popularize science and robotics among Estonian schoolchildren.

## Kasutatud kirjandus

1. MTÜ Robootika, Kooliroboti projekt [<http://www.robootika.ee/lego/projekt/>] viimati vaadatud 15.04.2011.
2. Vernier Software & Technology. [<http://www.vernier.com/>] viimati vaadatud 03.04.2011.
3. HiTechnic Products. [<http://www.hitechnic.com/>] viimati vaadatud 23.05.2011.
4. Mindsensors. [<http://www.mindsensors.com/>] viimati vaadatud 23.05.2011.
5. Codatex [[http://www.codatex.com/index.php?en\\_LEGO\\_MINDSTORMS\\_NXT](http://www.codatex.com/index.php?en_LEGO_MINDSTORMS_NXT)] viimati vaadatud 19.05.2011
6. Wikipedia, Prototype [<http://en.wikipedia.org/wiki/Prototype>] viimati vaadatud 28.12.2010
7. Wikipedia, Electronics prototyping [[http://en.wikipedia.org/wiki/Prototype#Electronics\\_prototyping](http://en.wikipedia.org/wiki/Prototype#Electronics_prototyping)] viimati vaadatud 28.12.2010
8. Wikipedia, Breadboard [<http://en.wikipedia.org/wiki/Breadboard>] viimati vaadatud 15.01.2011
9. Ldengineering.org, Solderless Breadboard [[http://www.ldengineering.org/wp-content/uploads/2010/11/solderless\\_breadboard\\_3001.jpg](http://www.ldengineering.org/wp-content/uploads/2010/11/solderless_breadboard_3001.jpg)] viimati vaadatud 15.01.2011
10. HiTechnic products – Prototyping Sensor Board [<http://www.hitechnic.com/cgi-bin/commerce.cgi?preadd=action&key=NPS1055>] viimati vaadatud 23.05.2011
11. HiTechnic, Solderable Prototype Sensor Board [<http://www.hitechnic.com/contents/media/Solderable-proto1.gif>] viimati vaadatud 15.01.2011
12. Techmind.org, Digital [<http://www.techmind.org/digital/>] viimati vaadatud 30.01.2011
13. Thermometrics.com, What is Thermistor? [<http://www.thermometrics.com/htmldocs/whatis.htm>] viimati vaadatud 23.05.2011
14. Wikipedia, Communications Protocol [[http://en.wikipedia.org/wiki/Communications\\_protocol](http://en.wikipedia.org/wiki/Communications_protocol)] viimati vaadatud 23.04.2011
15. Martin Loginov, LEGO MINDSTORMS NXT'ga ühilduva anduri NXTCam tutvustus ja ülesanded, peatükk 1.2.4 I<sup>2</sup>C Suhtlusprotokoll [<http://raju.cs.ut.ee/loputood/2010/martin-loginov/>] viimati vaadatud 23.04.2011

16. Siim Kallason, LEGO MINDSTORMS NXT Andurid  
<http://raju.cs.ut.ee/loputood/2009/siim-kallson/> viimati vaadatud 23.04.2011
17. Wikipedia, Microcontroller [\[http://en.wikipedia.org/wiki/Microcontroller\]](http://en.wikipedia.org/wiki/Microcontroller) viimati vaadatud 15.03.2011
18. Microchip, PIC16LF819-I/SS  
[\[http://media.digikey.com/photos/Microchip%20Tech%20Photos/150-20-SSOP.jpg\]](http://media.digikey.com/photos/Microchip%20Tech%20Photos/150-20-SSOP.jpg)  
viimati vaadatud 15.03.2011
19. Wikipedia, SSOP [\[http://en.wikipedia.org/wiki/Shrink\\_Small-Outline\\_Package\]](http://en.wikipedia.org/wiki/Shrink_Small-Outline_Package)  
viimati vaadatud 15.03.2011
20. Microchip, PIC16LF819-I/SS datasheet  
[\[http://www.chipcatalog.com/Microchip/PIC16LF819-I-SS.htm\]](http://www.chipcatalog.com/Microchip/PIC16LF819-I-SS.htm) viimati vaadatud 15.03.2011
21. Not eXactly C [\[http://bricxcc.sourceforge.net/nbc/\]](http://bricxcc.sourceforge.net/nbc/) viimati vaadatud 18.05.2011
22. Antistatic wrist strap  
[\[http://images.hightspeedbackbone.net/skuimages/large/ULT31418\\_main\\_image001\\_im.jpg\]](http://images.hightspeedbackbone.net/skuimages/large/ULT31418_main_image001_im.jpg) viimati vaadatud 20.05.2011
23. Extremen NXT, Pow[\[http://www.extremenxt.com/pow.zip\]](http://www.extremenxt.com/pow.zip) viimati vaadatud 23.04.2011
24. Team Hassenplug, BitLogic [\[http://www.teamhassenplug.org/NXT/BitLogic.zip\]](http://www.teamhassenplug.org/NXT/BitLogic.zip)  
viimati vaadatud 24.05.2011
25. Robootika.com, Kiiruste testimine [\[http://robootika.com/post/7-tund-NXT-kiiruse-testimine.aspx\]](http://robootika.com/post/7-tund-NXT-kiiruse-testimine.aspx) viimati vaadatud 14.04.2011
26. NXC API [\[http://bricxcc.sourceforge.net/nbc/nxcdoc/nxcapi/main.html\]](http://bricxcc.sourceforge.net/nbc/nxcdoc/nxcapi/main.html) viimati vaadatud 28.05.2011
27. NXC API, Lowspeed module  
[\[http://bricxcc.sourceforge.net/nbc/nxcdoc/nxcapi/group\\_low\\_speed\\_module\\_functions.html\]](http://bricxcc.sourceforge.net/nbc/nxcdoc/nxcapi/group_low_speed_module_functions.html) viimati vaadatud 28.05.2011
28. NXC API, I2CBytes  
[\[http://bricxcc.sourceforge.net/nbc/nxcdoc/nxcapi/group\\_low\\_speed\\_module\\_functions\\_ga085c21caa6674a6066c1cf67b7ef5374.html#ga085c21caa6674a6066c1cf67b7ef5374\]](http://bricxcc.sourceforge.net/nbc/nxcdoc/nxcapi/group_low_speed_module_functions_ga085c21caa6674a6066c1cf67b7ef5374.html#ga085c21caa6674a6066c1cf67b7ef5374) viimati vaadatud 28.05.2011
29. NXC API, I2CCheckStatus  
[\[http://bricxcc.sourceforge.net/nbc/nxcdoc/nxcapi/group\\_low\\_speed\\_module\\_functions.html\]](http://bricxcc.sourceforge.net/nbc/nxcdoc/nxcapi/group_low_speed_module_functions.html)

- [ons\\_gaa9aa51f9a9c4483e8490de8f6543b2c2.html#gaa9aa51f9a9c4483e8490de8f6543b2c2](http://bricxcc.sourceforge.net/nbc/nxcdoc/nxcapi/group_array_functions_ga49d48de55d817e96ed132eadc21260eb.html#ga49d48de55d817e96ed132eadc21260eb)] viimati vaadatud 28.05.2011
30. NXC API, ArrayInit  
[[http://bricxcc.sourceforge.net/nbc/nxcdoc/nxcapi/group\\_array\\_functions\\_ga49d48de55d817e96ed132eadc21260eb.html#ga49d48de55d817e96ed132eadc21260eb](http://bricxcc.sourceforge.net/nbc/nxcdoc/nxcapi/group_array_functions_ga49d48de55d817e96ed132eadc21260eb.html#ga49d48de55d817e96ed132eadc21260eb)] viimati vaadatud 24.05.2011
31. NXC API, Wait  
[[http://bricxcc.sourceforge.net/nbc/nxcdoc/nxcapi/group\\_command\\_module\\_functions\\_ga01e64d2250db0e5b41486e316228983f.html#ga01e64d2250db0e5b41486e316228983f](http://bricxcc.sourceforge.net/nbc/nxcdoc/nxcapi/group_command_module_functions_ga01e64d2250db0e5b41486e316228983f.html#ga01e64d2250db0e5b41486e316228983f)] viimati vaadatud 24.05.2011
32. Youtube, Custom Knight rider intro 1 – classic  
[<http://www.youtube.com/watch?v=Mo8Qls0HnWo>] viimati vaadatud 30.05.2011
33. Robootika.com, HiTechnic prototüüpimislaud [<http://robootika.com/post/Juhend-HiTechnic-prototuupimislaud.aspx>] viimati vaadatud 15.03.2011
34. Wikipedia – Light-emitting diode [[http://en.wikipedia.org/wiki/Light-emitting\\_diode#Colors\\_and\\_materials](http://en.wikipedia.org/wiki/Light-emitting_diode#Colors_and_materials)] viimati vaadatud 24.04.2011
35. Hyperphysics, Ohm law [<http://hyperphysics.phy-astr.gsu.edu/hbase/electric/ohmlaw.html>]
36. Martin Loginov, LEGO MINDSTORMS NXT'ga ühilduva anduri NXTCam tutvustus ja ülesanded, peatükk 2.4.1 Valgustingimused  
[<http://raju.cs.ut.ee/loputood/2010/martin-loginov/>] viimati vaadatud 23.04.2011
37. HiTechnic, Experimenters Kit Handbook [[http://www.hitechnic.com/file.php?f=438-HiTechnic Experimenters Kit Release v1.0.zip](http://www.hitechnic.com/file.php?f=438-HiTechnic_Experimenters_Kit_Release_v1.0.zip)] viimati vaadatud 24.04.2011
38. NXC API, Random  
[[http://bricxcc.sourceforge.net/nbc/nxcdoc/nxcapi/ex\\_random\\_8nxc-example.html](http://bricxcc.sourceforge.net/nbc/nxcdoc/nxcapi/ex_random_8nxc-example.html)] viimati vaadatud 28.05.2011
39. NXC API, Current Tick  
[[http://bricxcc.sourceforge.net/nbc/nxcdoc/nxcapi/ex\\_current\\_tick\\_8nxc-example.html](http://bricxcc.sourceforge.net/nbc/nxcdoc/nxcapi/ex_current_tick_8nxc-example.html)] viimati vaadatud 28.05.2011
40. NXC API, TextOut  
[[http://bricxcc.sourceforge.net/nbc/nxcdoc/nxcapi/ex\\_text\\_out\\_8nxc-example.html](http://bricxcc.sourceforge.net/nbc/nxcdoc/nxcapi/ex_text_out_8nxc-example.html)] viimati vaadatud 28.05.2011

# Lisad

## Lisa 1 – Ülesannete lahendused

Tabelis 3 on toodud käesoleva bakalaureusetöö 3. osas kirjeldatud ülesannete lahendusfailid. Tegemist on NXT-G (\*.rbt) ja NXC (\*.nxc) programmifailidega.

Faili nimi	Ülesande number ja nimi
yl1.rbt	Ülesanne 1 - Vilkuv LED NXT-G keeles
yl1.nxc	Ülesanne 1 - Vilkuv LED NXC keeles
y2.rbt	Ülesanne 2 - KITT'i tuled NXT-G keeles
yl2.nxc	Ülesanne 2 - KITT'i tuled NXC keeles
yl3.rbt	Ülesanne 3 - Valguse andur
yl3.nxc	Ülesanne 3 - Valguse andur
yl4.rbt	Ülesanne 4 - Reaktsiooni aja mõõtmine
yl4.nxc	Ülesanne 4 - Reaktsiooni aja mõõtmine

**Tabel 3.** CD plaadi peal olevad ülesannete lahendusfailid